# Clustering functional data using wavelets

Anestis Antoniadis, Xavier Brossat, Jairo Cugliari, Jean-Michel Poggi

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Clustering functional data using wavelets*

Anestis Antoniadis — Xavier Brossat — Jairo Cugliari — Jean-Michel Poggi

**N° 7515**

Mai 2011

Optimization, Learning and Statistical Methods


ISSN 0249-6399 ISRN INRIA/RR--7515--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SACLAY - ÎLE-DE-FRANCE

# Clustering functional data using wavelets

Anestis Antoniadis* , Xavier Brossat† , Jairo Cugliari‡ ,
Jean-Michel Poggi§

Theme : Optimization, Learning and Statistical Methods
Applied Mathematics, Computation and Simulation
Équipe-Projet SELECT

Rapport de recherche n° 7515 — Mai 2011 — 32 pages

**Abstract:** We present two methods for detecting patterns and clusters in high dimensional time-dependent functional data. Our methods are based on wavelet-based similarity measures, since wavelets are well suited for identifying highly discriminant local time and scale features. The multiresolution aspect of the wavelet transform provides a time-scale decomposition of the signals allowing to visualize and to cluster the functional data into homogeneous groups. For each input function, through its empirical orthogonal wavelet transform the first method uses the distribution of energy across scales generate a handy number of features that can be sufficient to still make the signals well distinguishable. Our new similarity measure combined with an efficient feature selection technique in the wavelet domain is then used within more or less classical clustering algorithms to effectively differentiate among high dimensional populations. The second method uses dissimilarity measures between the whole time-scale representations and are based on wavelet-coherence tools. The clustering is then performed using a k-centroid algorithm starting from these dissimilarities. Practical performance of these methods that jointly designs both the feature selection in the wavelet domain and the classification distance is demonstrated through simulations as well as daily profiles of the French electricity power demand.

**Key-words:** functional data, clustering, electricity power demand, wavelets

* Université Joseph Fourier, Laboratoire LJK, Tour IRMA, BP53, 38041 Grenoble Cedex 9, France anestis.antoniadis@imag.fr

† EDF R&D, 1 avenue du Général de Gaulle, 92141 Clamart Cedex, France xavier.brossat@edf.fr

‡ EDF R&D; Université Paris-Sud, France jairo.cugliari@math.u-psud.fr

§ Université Paris 5 Descartes; Université Paris-Sud, France jean-michel.poggi@math.u-psud.fr

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

# Classification non supervise des donnes fonctionnelles  l'aide d'ondelettes

**Résumé :** Nous prsentons deux mthodes de dtection des clusters dans des donnes fonctionnelles avec une structure temporelle. Nos mthodes sont bases sur des mesures de similarit fonds dans la dcomposition en ondelettes, car cette dcomposition est bien adapte pour identifier des caractristiques localises en temps et chelles. L'aspect multi-rsolution de la transforme en ondelettes permet une dcomposition des signaux en temps et chelle permettant de visualiser et de regrouper les donnes fonctionnelles en groupes homognes. La premire mthode permet d'obtenir,  travers la transform orthogonal empirique en ondelettes, la distribution de l'nergie  travers les chelles et ainsi de gnrer un faible nombre d'attributs encore capables de bien distinguer les signaux. La nouvelle mesure de similarit associe  une technique efficace de slection des variables dans le domaine des ondelettes est ensuite utilise dans des algorithmes de clustering classiques. La deuxime mthode utilise des mesures de dissimilitude entre les reprsentations temps-chelle bass sur la notion de cohrence en ondelettes. Nous utilisons un algorithme de k-centroides pour obtenir les groupes  partir de ces dissimilitudes. La performance pratique de nos mthodes est dmontre  travers de simulations et des profiles de consommation journalire d'nergie lectrique en France.

**Mots-clés :** donnes fonctionnelles, classification non supervise, demande d'nergie lectrique, ondelettes

# 1 Introduction

In different fields of applications, explanatory variables are not standard multivariate observations, but are functions observed either discretely or continuously. Ramsay and Dalzell (1991) gave the name "functional data analysis" to the analysis of data of this kind. As evidenced in the work by Ramsay and Silverman (1997, 2002) (see also Ferraty and Vieu (2006)), a growing interest is notable in investigating the dependence relationships between complex functional data such as curves, spectra, time series or more generally signals. Functional data often arise from measurements on fine time grids, and if the sampling grid is sufficiently dense, the resulting data may be viewed as a sample of curves. These curves may vary in shape, both in amplitude and phase. Typical examples involving functional data can be found when studying the forecasting of electricity consumption, temporal gene expression analysis or ozone concentration in environmental studies to cite only a few.

Given a sample of curves, an important task is to search for homogeneous subgroups of curves using clustering and classification. Clustering is one of the most frequently used data mining techniques, which is an unsupervised learning process for partitioning a data set into sub-groups so that the instances within a group are similar to each other and are very dissimilar to the instances of other groups. In a functional context clustering helps to identify representative curve patterns and individuals who are very likely involved in the same or similar processes. Recently, several functional clustering methods have been developed such as variants of the k-means method (Tarpey and Kinateder (2003); Tarpey (2007); Cuesta-Albertos and Fraiman (2007)) and clustering after transformation and smoothing (CATS) (Serban and Wasserman (2004)) to model-based procedures, such as clustering sparsely sampled functional data (James and Sugar (2003)) or mixed effects modeling approach using B-splines (Luan and Li (2003)) that mostly concentrate on curves exhibiting a regular behaviour.

Our interest in time series of curves is motivated by an application in forecasting a functional time series when the most recent curve is observed. This situation arises frequently when a seasonal univariate continuous time series is segmented into consecutive segments, for example days, and treated as a discrete time series of functions. The idea of forming a functional valued discrete time series from segmentation of a seasonal univariate time series has been introduced by Bosq (1991). Suppose one observes a square integrable continuous time stochastic process $X = (X(t), t \in \mathbb{R})$ over the interval $[0, T]$, $T > 0$ at a relatively high sampling frequency. The commonly used approach is to divide the interval $[0, T]$ into sub-intervals $[(l-1)\delta, l\delta]$, $l = 1, \ldots, n$ with $\delta = T/n$, and to consider the functional-valued discrete time stochastic process $Z = (Z_i, i \in \mathbb{N})$, defined by

$$Z_i(t) = X(t + (i-1)\delta), \qquad i \in \mathbb{N} \quad \forall t \in [0, \delta). \tag{1}$$

The random functions $Z_i$ thus obtained, while exhibiting a possibly non-stationary behavior within each continuous time subinterval, form a functional discrete times series that is usually assumed to be stationary. Such a procedure allows to handle seasonal variation of size $\delta$ in a natural way. This set-up has been used for prediction when ones consider a Hilbert-valued discrete time stationary autoregressive processes (see Bosq (1991); Besse and Cardot (1996); Pumo (1992); Antoniadis and Sapatinas (2003)) or for more general continuous-

time processes (see Antoniadis *et al.* (2006)). However, as already noticed above, for many functional data the segmentation into subintervals of length $\delta$ may not suffice to make reasonable the stationary hypothesis of the resulting segments, that is the key for efficient prediction. For instance, in modeling the electrical power demand process the seasonal effect of temperature and the calendar configuration strongly affects the mean level and the shape of the daily load demand profile. *Recognizing this, our aim is therefore to propose a clustering technique that clusters the functional valued (discrete) times series segments into groups that may be considered as stationary so that in each group more or less standard functional prediction procedures such as the one cited above can be applied.*

We will apply the methodology focusing on EDF's (*Électricité de France*[1]) national power demand for a year. This is essentially a continuous process even though we only count with discrete records sampled at 30 minutes for the whole year. Some of the facts associated to the electricity power demand induce to think that the process is not stationary. We will construct a functional data set by splitting the continuous process as in equation (1) where the parameter $\delta$ will be a day.

Although slicing an univariate time series produces functional data, we do not observe the whole segments but a sample of the values at some time points. One could then use a vector representation of each observation. For example Wang *et al.* (2008) proposed to measure the distance between observations through the high dimensional multivariate distribution of all sampled time points along each curve. This approach does not exploit the eventually potential information of correlations between points of a single curve. To avoid this, many authors have clustered the coefficients of a suitable basis representation of functions. Since the analyzed curves are infinite-dimensional and temporal-structured, one projects each curve over an appropriate basis of the functional space to extract specific features form the data which are then used as inputs for clustering or classification. One may cite for example Abraham *et al.* (2003) where the authors proposed to cluster the spline fit coefficients of the curves using $k-$means, or James and Sugar (2003) that use a spline decomposition specially adapted for sparsely sample functional data. Nevertheless, attention must be paid to the chosen basis because this operation involves linear transformation of data that may not be invariant for the clustering technique (see Tarpey (2007)). Splines are often used to describe functions with a certain degree of regularity. However, we will be working with curves like in Figure 7 that may present quite irregular paths. We chose to work with wavelets because of their good approximations properties on sample paths that might be rather irregular. One may note here that similar methodologies relying upon wavelet decompositions for clustering or classifying time series have been developed in the literature (e.g. see Pittner *et al.* (1999) and Ray & Mallick (2006)).

Wavelets offer an excellent framework when data are not stationary. For example, in Gurley *et al.* (2003) the wavelet transform is used to develop the concept of wavelet-coherence that describes the local correlation of the time-scale representation of two functions. Grinsted *et al.* (2004) proved that this concept is convenient for clustering geophysical time series. Another example supporting such a fact is the work by Quiroga *et al.* (2004) which uses wavelets to detect and cluster spikes on neural activity. Motivated by this and the fact

[1] `http://www.edf.com`

that the wavelet transform has the property of time-frequency localization of the time series, we propose hereafter a time-series feature extraction algorithm using orthogonal wavelets for automatically choosing feature dimensionality for clustering. We also study some more complex alternatives using the wavelet-coherence concept in sake of better exploiting the well localized information of the wavelet transform.

The rest of the paper is organized as follows. Section 2 is a reminder on multiresolution analysis and introduces the basis supporting our feature extraction algorithm by means of the energy operator. Following wavelet analysis we cluster the functional data using the extracted features in Section 3. Our first clustering algorithm uses $k$-means as unsupervised learning routine. We test the proposed method in Section 4 on simulated and real data. Section 5 presents a more sophisticated method for clustering functional data using a more specific dissimilarity measure. Finally, we conclude the paper by summarizing the main contributions and perspectives in Section 6.

# 2 Feature extraction with wavelets

In this section we first introduce some basic ideas of the wavelet analysis before introducing more specific material: the energy contributions of the scale levels of the wavelet transform which are the key tools for future clustering. More details about wavelets and wavelet transforms can be found for example in Mallat (1999).

We will consider a probability space $(\Omega, \mathcal{F}, P)$ where we define a function-valued random variable $Z : \Omega \to \mathcal{H}$, where $\mathcal{H}$ is a (real) separable Hilbert space (e.g. $\mathcal{H} = L^2(\mathcal{T})$ the space of squared-integrable functions on $\mathcal{T} = [0, 1)$ (finite energy signals) or $\mathcal{H} = W_2^s(\mathcal{T})$ the Sobolev space of $s$-smooth function on $\mathcal{T}$, with integer regularity index $s \geq 1$) endowed with the Hilbert inner product $< ., . >_{\mathcal{H}}$ and the Hilbert norm $\|.\|_{\mathcal{H}}$.

## 2.1 Wavelet transform

A wavelet transform (WT for short) is a domain transform technique for hierarchical decomposing finite energy signals. It allows a real valued function to be described in terms of an approximation of the original function, plus a set of details that range from coarse to fine. The property of wavelets is that the broad trend of the input function is captured in the approximation part, while the localized changes are kept in the detail components. For short, a wavelet is a smooth and quickly vanishing oscillating function with good localization properties in both frequency and time. This is suitable for approximating curves that contain localized structures. A compactly supported WT uses a orthonormal basis of waveforms derived from scaling (i.e. dilating or stretching) and translating a compactly supported scaling function $\widetilde{\phi}$ and a compactly supported mother wavelet $\widetilde{\psi}$. We consider periodized wavelets in order to work over the interval $[0, 1]$, denoting by

$$\phi(t) = \sum_{l \in \mathbb{Z}} \widetilde{\phi}(t - l) \quad \text{and} \quad \psi(t) = \sum_{l \in \mathbb{Z}} \widetilde{\psi}(t - l), \quad \text{for} \quad t \in [0, 1],$$

the periodized scaling function and wavelet, that we dilate or stretch and translate

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k), \quad \psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k).$$

For any $j_0 \geq 0$, the collection

$$\{\phi_{j_0,k}, k = 0, 1, \ldots, 2^{j_0} - 1; \psi_{j,k}, j \geq j_0, k = 0, 1, \ldots, 2^j - 1\},$$

is an orthonormal basis of $\mathcal{H}$. Thus, any function $z \in \mathcal{H}$ can then be decomposed in terms of this orthogonal basis as

$$z(t) = \sum_{k=0}^{2^{j_0}-1} c_{j_0,k}\phi_{j_0,k}(t) + \sum_{j=j_0}^{\infty}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(t), \tag{2}$$

where $c_{j,k}$ and $d_{j,k}$ are called respectively the scale and the wavelet coefficients of $z$ at the position $k$ of the scale $j$ defined as

$$c_{j,k} =< z, \phi_{j,k} >_{\mathcal{H}}, \quad d_{j,k} =< z, \psi_{j,k} >_{\mathcal{H}} .$$

To efficiently calculate the WT, Mallat introduced the notion of mutiresolution analysis of $\mathcal{H}$ (MRA) and designed a family of fast algorithms (see Mallat (1999)).

With MRA, the first term at the right hand side of (2) can be viewed as a smooth approximation of the function $z$ at a resolution level $j_0$. The second term is the approximation error. It is composed by the aggregation of the details at scales $j \geq j_0$. These two components, approximation and details, can be viewed as a low frequency (smooth) nonstationary part and a component that keeps the time-localized details at higher scales. The distinction between the smooth part and the details is determined by the resolution $j_0$, that is the scale below which the details of a signal cannot be distinguished. We will focus our attention on the finer details, i.e. on the information at the scales $\{j : j \geq j_0\}$.

From a practical of view, each function is usually observed on a fine time sampling grid of size $N$. In the sequel we will be interested in input signals of length $N = 2^J$ for some integer $J$. If $N$ is not a power of 2, one may interpolate data to the nearest $J$ with $2^{J-1} < N < 2^J$. We have already seen that an advantage of the nested structure of a multiresolution analysis is that it leads to an efficient tree-structured algorithm for the decomposition of functions in $V_J$ (Mallat (1989)) for which the coefficients $< z, \phi_{J,k} >_{\mathcal{H}}$ are given and that allows to derive the coefficients of the Discrete Wavelet Transform (DWT). However, when a function $z$ is given in sampled form there is no general method for deriving the coefficients $< z, \phi_{N,k} >_{\mathcal{H}}$ and one has to approximate the projection $P_{V_J}$ by some operator $\Pi_J$ in terms of the sampled values $\mathbf{z} = \{z(t_l) : l = 0, \ldots, N-1\}$ of $z$. For regular enough wavelets, such an approximation is highly efficient (see Antoniadis (1994)) and justifies the following.

Denote by $\mathbf{z} = \{z(t_l) : l = 0, \ldots, N-1\}$ the finite dimensional sample of the function $z$. For the particular level of granularity given by the size $N$ of the sampling grid, one rewrites the approximation $\Pi_J$ of the projection $P_{V_J}$ of $z$ in (2) using the truncation imposed by the $2^J$ points and the coarser approximation level $j_0 = 0$, as:

$$\widetilde{z}_J(t) = c_0\phi_{0,0}(t) + \sum_{j=0}^{J-1}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(t), \tag{3}$$

where $c_0$ and $d_{j,k}$ are now denoting the empirical wavelet coefficients derived from applying the DWT on the sampled values. Hence, for a chosen, regular enough, wavelet $\psi$ and a coarse resolution $j_0 = 0$, one may define the DWT operator:

$$W_\psi : \mathbb{R}^N \to \mathbb{R}^N, \quad \mathbf{z} \mapsto (\mathbf{d}_0, \ldots, \mathbf{d}_{J-1}, c_0)\,,$$

with $\mathbf{d}_j = \{d_{j,0}, \ldots, d_{j,2^j-1}\}$. Since the DWT operator is based on an $L_2$-orthonormal basis decomposition, Parseval's theorem states that the energy of a square integrable signal is preserved under the orthogonal wavelet transform:

$$\|\mathbf{z}\|_2^2 = c_0^2 + \sum_{j=0}^{J-1} \sum_{k=0}^{2^j-1} d_{j,k}^2 = c_0^2 + \sum_{j=0}^{J-1} \|\mathbf{d}_j\|_2^2. \tag{4}$$

Hence, the global energy $\|\mathbf{z}\|_2^2$ of $\mathbf{z}$ is broken down into a few energy components. The representation (4) is in fact composed by the components of the discrete wavelet scalogram as defined in Arino, Morettin and Vidakovic (2004) and may be seen as the (DWT) analogue of the well-known periodogram from the spectral analysis of time series. Just as the periodogram produces an ANOVA decomposition of the energy of a signal into different Fourier frequencies, the scalogram decomposes the energy into "level components". Since $N = 2^J$ no more than $J$ such levels can be defined. After removing from each continuous time series slowly varying trends and eventual periodicities in time by disregarding the approximation coefficient $c_0$, the scalogram components indicate at which levels of resolution the energy of the observed function is concentrated. A relatively smooth function will have most of its energy concentrated in large-scale levels, yielding a scalogram that is large for small $j$ and small for large $j$. A function with a lot of high frequency oscillations will have a large portion of its energy concentrated in high resolution wavelet coefficients. Therefore the way these energies components are distributed and contribute to the global energy of a signal is the key fact that we are going to exploit to generate a handy number of features that are going to be used for clustering.

The image of the DWT operator applied on the column vector $\mathbf{z}$ of dimension $N = 2^J$ may be written in matrix form as:

$$\mathbf{W} = \mathcal{W}\mathbf{z},$$

where $\mathcal{W}$ is a $N$ by $N$ square matrix defining the DWT and satisfying $\mathcal{W}'\mathcal{W} = I_N$ (see Percival & Walden (2006, chap. 4)), and $\mathbf{W}$ is a column vector of length $N$ with

$$\mathbf{W} = (\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{J-1}, c_0)',$$

where $W'$ denotes the transpose of $W$. It is easy to see that if we consider a vector $\mathbf{x} = a + b\mathbf{z}$ with $a, b \in \mathbb{R}$, then the wavelet coefficients of the DWT of $\mathbf{x}$ are obtained from those of the $\mathbf{z}$ as:

$$(b\mathbf{d}_0, b\mathbf{d}_1, \ldots, b\mathbf{d}_{J-1}, a + bc_0)'. \tag{5}$$

## 2.2 Absolute and relative contributions

We just have seen that DWT coefficients describe properties of functions both at various locations and at various time granularities. Each time granularity here

refers to the level of detail that can be captured by DWT. This is therefore the reason of choosing the DWT as a representation scheme in our previous section to compare the shapes of curves for clustering. The energy $\mathcal{E}_z = \|z\|^2_{\mathcal{H}}$ of the time series $z$ via decomposition (4) is equal to the sum of the energy of its wavelet coefficients distributed across scales

$$\mathcal{E}_z \approx \|\mathbf{z}\|_2^2 = c_0^2 + \sum_{j=0}^{J-1} \|\mathbf{d}_j\|_2^2, \tag{6}$$

the approximation (denoted informally by $\approx$) holding because of the truncation at scale $J$ for the wavelet expansion of $z$, discarding finer scales. If we consider $\mathbf{z}$ as the difference between two sampled curves, (6) justifies using the energy decomposition of wavelet coefficients for computing squared Euclidean distances between two series. However, when interested to see how the energy of wavelet coefficients is distributed across scales, other distance functions on DWT decompositions may be more appropriate for measuring the dissimilarity between two series.

In what follows, we define for $j = 0, \ldots, J-1$ the absolute and relative contributions of the scale $j$ to the global energy of the centred function respectively as

$$\text{cont}_j = \|\mathbf{d}_j\|_2^2, \qquad \text{rel}_j = \frac{\text{cont}_j}{\sum_{j=0}^{J-1} \text{cont}_j}, \qquad \forall j = 0, \ldots, J-1. \tag{7}$$

We call these representations: the absolute contribution (AC) and the relative contribution (RC). We will therefore characterize each time series by the vector of its energy contributions or its relative contributions in order to define an appropriate measure of similarity that is going to be used for clustering. Note that in both of these choices of representation we leave out the eventual mean level differences of the time series since we do not make any use of the approximation term $c_0$ in their definition. Indeed, when within a same similarity class, the discrete time function valued processes $Z$ are considered as stationary and therefore the scaling coefficient $c_0$ does not have any discriminative power, that's why we use only details after 0 in defining our distance. Now in order to compare two paths and since they are zero-mean, our distance is more relevant to measure a difference on how their energies differ across scales. In our ultimate goal to predict the future behaviour of the discrete time functional valued process $Z$, collecting the series of scaling coefficients for each segment $Z_i$, $i = 1, \ldots, n$ one ends up with a real valued process of approximation coefficients. For a sequence of such univariate approximation coefficients, say $\{c_{0,1}, \ldots, c_{0,n}\}$, more or less classical time series models can be used to predict the next coefficient $c_{0,n+1}$. Using this fact, the considered dissimilarities computed throughout the representations will be invariant under vertical shifts of the curves. Moreover, using RC implies fixing the energy of the curves to one. Hence no difference in amplitude can be detected.

# 3 A $k-$means like functional clustering procedure

We have presented a way to represent the infinite-dimensional original objects in $J$ features that summarize the time evolution of the curves at different scales. We will now see how we use the information that we have coded to effectively cluster it.

This section starts with a brief review of some recent and highly efficient developments on feature selection and on the choice of the number of cluster.

## 3.1 Feature selection

Nothing warranties that all the extracted features are relevant to discover the cluster structure. Analogously with regression analysis, a feature selection step can be performed to detect the significant ones.

Feature extraction and features selection have really different aims. Whether the former creates some new information from existing objects, the latter only selects a subset of existing features. This selection reduces the computational time of the algorithm and helps to avoid an unsatisfactory and unstable clustering. Another important advantage of using a feature selection algorithm is that the reduced number of features helps to better understand the cluster output. In our case, the number of features depends on the number of sampling points of the acquired data. For $N$ points, the number of features is $J = \log_2(N)$ that can be large. Moreover, since we are interested in the energy decomposition across scales, potentially several scales will not be informative for the cluster structure. Besides this, the feature selection algorithm aims to reduce (or eliminate) the presence of nonsignificant variables and a possible redundant information that could hide the cluster structure.

The absence of class labels on unsupervised learning rends particularly difficult the feature selection task. Besides, this task is intricately connected with the determination of the number of cluster. Thus, it is desirable to conduce both of them simultaneously. A recent comparative study (Steinley & Brusco (2008)) evaluate the performance of eight feature selection algorithms on simulated data sets. The compared algorithms covers model-based approaches (e.g. Law *et al.* (2004) which allow the simultaneous detection of groups and features) and nonparametric ones. The same authors proposed in Steinley & Brusco (2008) an algorithm that combines a variable transformation with a variable selection technique. The variable transformation introduces a variance-to-ratio weighting that looks for placing the variables on the same scale while preserving their ability to reveal cluster structures. Then, the transformed variables are used to construct an index of clusterability that serves to screen the variables that do not reveal information about the cluster structure (which is useful when working with large data sets with many masking features). Then, for the remaining variables an exhaustive evaluation of the feasible subsets of variables is done. For each subset size $s$, a best set of variables is obtained in terms of the largest proportion of explained variation VAF($s$) from the clustering. Note that VAF decreases monotonically as a function of $s$. Finally, the solution of the algorithm

is the subset of variables that maximizes the following ratio

$$\frac{\mathrm{VAF}(s) - \mathrm{VAF}(s+1)}{\mathrm{VAF}(s-1) - \mathrm{VAF}(s)},$$

where the heuristic is that for the right number of features, say $s^*$, adding one extra variable produces a larger degradation on the VAF than the one produced when passing from $s^* - 1$ to $s^*$.

## 3.2 Determination of the number of clusters

One of the most difficult task in clustering is the determination of the number of clusters $K$. Even if some statistical support can be given to achieve this task, usually the knowledge on the particular application helps on the choice. In the classical case, i.e. not the functional one, a lot of data-driven strategies can be defined. The first one by inspecting basically the within-cluster dissimilarity as a function of $K$. Many heuristics have been proposed trying to find a "kink" in the corresponding plot.

A more formal argument has been proposed by Tibshirani *et al.* (2001) by comparing, using the gap statistic, the logarithm of the empirical within-cluster dissimilarity and the corresponding one for uniformly distributed data. Slight modifications have been proposed to the original argument, see for instance Ye (2007).

Another point of view useful to determine the number of clusters comes from model-based clustering. The idea is to fit a Gaussian mixture model to data and identify clusters as mixture components. The number of clusters is usually obtained using the well-known BIC criterion. All the above mentioned strategies seem to perform well when data do come from a mixture model but can perform poorly when the situation is more confused and fuzzy.

For determining the number of clusters James and Sugar (2003) proposed an information theoretic approach. They consider the transformed distortion curve $d_K^{-p/2}$, a kind of average Mahalanobis distance between data and the set of cluster centers $C$ as a function of $K$. Jumps in the associated plot allow to select sensible values for $K$ while the largest one can be the best choice for a mixture of $p$-dimensional distributions with common covariance. An asymptotic analysis (as $p$ goes to infinity) states that, when the number of clusters used is smaller than the correct number (when any), then the transformed distortion remains close to zero, before jumping suddenly and increasing linearly. Then, detecting a jump in the transformed distortion curve is equivalent to detecting the number of clusters $K$.

## 3.3 The actual procedure

We can now sum up the actual strategy for clustering that we will use in the first part of the paper in the following steps:

**0. Data preprocessing.** Approximate sample paths of $z_1(t), \ldots, z_n(t)$ by the truncated wavelet series (see (3)) at the scale $J$ from sampled data $\mathbf{z}_1, \ldots, \mathbf{z}_n$.

**1. Feature extraction.** Compute either the energetic components using absolute contribution (AC) or relative contribution (RC). If using the latter, transform the obtained vector using the logit transformation.