ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Simultaneous Clustering and Segmentation for Functional Data

Bernard Hugueney[1], Georges Hébrail[2], Yves Lechevallier[3] and Fabrice Rossi[2]

1– LAMSADE, Université Paris Dauphine,
Place du Maréchal de Lattre de Tassigny, 75016 Paris – France

2– Institut TELECOM, TELECOM ParisTech, LTCI - UMR CNRS 5141
46, rue Barrault, 75013 Paris – France

3– Projet AxIS, INRIA, Domaine de Voluceau, Rocquencourt, B.P. 105
78153 Le Chesnay Cedex – France

Abstract. We propose in this paper an exploratory analysis algorithm for functional data. The method partitions a set of functions into K clusters and represents each cluster by a piecewise constant prototype. The total number of segments in the prototypes, P, is chosen by the user and optimally distributed into the clusters via two dynamic programming algorithms.

1 Introduction

Functional Data Analysis [8] addresses problems in which the observations are described by functions rather than finite dimensional vectors. A well known real world example of such data is given by spectrometry in which each object is analysed via one spectrum, that is a function which maps wavelengths to absorbance values. Online monitoring of hardware is also a good example of such data: each object is described by several time series associated to physical quantities monitored at specified sampling rate.

We focus in this paper on the exploratory analysis of a set of curves (or time series). The main idea is to provide the analyst with a summary of the set with a manageable complexity. A classical solution for multivariate data consists in using a prototype based clustering approach: each cluster is summarized by its prototype. Standard clustering methods such as K-means and Self Organizing Map have been adapted to functional data and could be used to implement this solution [1, 9]. Another possibility comes from the symbolization approaches in which a time series is represented by a sequence of symbols [4]: a piecewise constant approximation of a time series is constructed via a segmentation of the time domain into contiguous intervals on which the series is represented by its average value, which can be turned into a label in a subsequent quantization step. When we are given a set of curves, an unique segmentation can be found in order to represent all the curves on a common piecewise constant basis (see [11] for an optimal solution). This was used as a preprocessing step in e.g. [10, 5].

We propose in this paper to merge the two approaches: we build a k-means like clustering of a set of functions in which each prototype is given by a piecewise constant function. Using dynamic programming, we obtain an optimal segmentation in each cluster while the number of segments used in each cluster is also optimally chosen with respect to a user specified total number segments. In other words, a set of functions is summarized via $K \times P$ real values, where K is the number of prototypes and P the total number of segments used to represent the prototypes.

The rest of this paper is organized as follows. Section 2 describes the dynamic programming algorithm used to obtain an optimal segmentation of the mean curve of a set of curves. Section 3 shows how this can be combined to a clustering algorithm with optimal resource allocation between clusters. Section 4 illustrates the method on a real world dataset.

2 Optimal segmentation

Let us first consider N functions $(s_i)_{i=1}^N$ sampled in M distinct points $(t_k)_{k=1}^M$ from the interval $[t_1, t_M]$ (points are assumed to be ordered). Our goal is to find a piecewise constant function defined on $[a, b]$ with minimal squared distance to all the functions. The complexity of the constant function is fixed by the number of intervals needed to define it. A complexity P corresponds to a partition of $[t_1, t_M]$ into P intervals $(I_p)_{p=1}^P$ and to P values $(a_p)_{p=1}^P$ which minimize

$$E\left((I_p)_{p=1}^P, (a_p)_{p=1}^P\right) = \sum_{i=1}^N \sum_{p=1}^P \sum_{t_k \in I_p} \left(s_i(t_k) - a_p\right)^2. \qquad (1)$$

The difficulty lies in the choice of the partition $(I_p)_{p=1}^P$, as, given this partition, the optimal values of the $(a_p)_{p=1}^P$ are equal to the $\mu_p = \frac{1}{N|I_p|}\sum_{i=1}^N \sum_{t_k \in I_p} s_i(t_k)$. To derive an efficient algorithm, we first note that the intervals $(I_p)_{p=1}^P$ define a partition of the series $t_1 < t_2 < \ldots < t_P$ with an ordering constraint: if $t_k \in I_p$ and $t_l \in I_p$, then $\{t_k, t_{k+1}, \ldots, t_l\} \subset I_p$. In fact, it is sufficient to define a partition of $(t_k)_{k=1}^M$ with ordering constraint to obtain a piecewise constant function, as we do not have information on the functions elsewhere than at the evaluation points. Given such a partition $(C_p)_{p=1}^P$, and using the optimal values of the $(a_p)_{p=1}^P$ defined above, the error to minimize is

$$E\left((C_p)_{p=1}^P\right) = \sum_{p=1}^P \sum_{i=1}^N \sum_{t_k \in C_p} \left(s_i(t_k) - \mu_p\right)^2 = \sum_{p=1}^P Q\left(C_p\right). \qquad (2)$$

The error measure is therefore additive over the partition. As pointed out in [6, 7], finding an optimal partition with ordering constraints with respect to an additive criterion can be done efficiently via a dynamic programming approach. In the particular case of constructing the optimal piecewise constant approximation of a single function, this was pointed out in [2] (see also [11] for the best basis problem in the case of a set of functions).

Let us define $F(k, j)$ as the quality measure (from equation (2)) of the best partition into j clusters with ordering constraint of $\{t_k, t_{k+1}, \ldots, t_M\}$. The basic idea is to compute $F(k, j)$ using $F(., j-1)$. Indeed the best partition into j

clusters is obtained by minimizing over l $Q(\{t_1,\ldots,t_l\})+F(l+1,j-1)$ for two reasons. First the partition is ordered and therefore we can ask C_1 to be equal to $\{t_1,\ldots,t_l\}$ for a certain l. Second, the quality measure is additive and therefore finding the best partition in j clusters with the constraint that $C_1=\{t_1,\ldots,t_l\}$ corresponds to finding the best partition of $\{t_{l+1},\ldots,t_M\}$ in $j-1$ clusters.

The algorithm proceeds as follows:

1. initialize $F(k,1)$ to $Q(\{t_k,t_{k+1},\ldots,t_M\})$
2. for j going from 2 to P:
   (a) for k going from 1 to $M-j+1$ compute
   $$F(k,j)=\min_{k\leq l\leq M-j+1} Q(\{t_k,\ldots,t_l\})+F(l+1,j-1)$$

We keep track of the wining index l for each $F(k,j)$: this allows to reconstruct the best partition when we have reached $F(1,P)$. Given all the $Q(\{t_k,\ldots,t_l\})$, this algorithm runs in $O(PM^2)$. The computation of the $Q(\{t_k,\ldots,t_l\})$ itself can be done quite efficiently. A naive approach leads to $O(NM^3)$, but a simple recursive formulation reduces the cost to $O(NM^2)$. Indeed we first compute the mean function $\mu(t_k)=\frac{1}{N}\sum_{i=1}^N s_i(t_k)$ in $O(NM)$. Then, we compute recursively the $\mu_{\{t_k,\ldots,t_l\}}$ in $O(M^2)$. Finally, $Q(\{t_k,\ldots,t_l\})$ are obtained via standard recursive variance calculation formulae for a total cost of $O(NM^2)$.

3 Clustering and resource allocations

The segmentation algorithm proposed in the previous Section can be easily embedded into a K means algorithm. The main idea is to optimize the following quality criterion

$$E\left((G_k)_{k=1}^K,(P_k)_{k=1}^K\right)=\sum_{k=1}^K\sum_{s_i\in G_k}\sum_{p=1}^{P_k}\sum_{t_l\in C_p^k}\left(s_i(t_l)-\mu_p^k\right)^2, \tag{3}$$

where $(G_k)_{k=1}^K$ is a partition of the functions into K clusters, $(P_k)_{k=1}^K$ are K integers such that $\sum_k P_k=P$ and $(C_p^k)_{p=1}^{P_k}$ are partitions which define the segmentations used by the prototypes. This is a clustered version of the measure proposed in equation (2). It can be optimized via the K means alternating scheme:

1. initialise the clusters with a random partition
2. for each cluster G_k find a piecewise constant prototype g_k with P_k segments
3. assign each function s_i to the best matching cluster, i.e., the cluster whose prototype has minimal squared distance to s_i
4. if the clusters have changed go back to step 2

This simple solution do not automatically allocate resources in an optimal way: the $(P_k)_{k=1}^{K}$ have to be given in advance. Fortunately, dynamic programming can be used again to remove this constraint. Indeed, for each $(P_k)_{k=1}^{K}$, the algorithm proposed in Section 2 leads to an optimal set of partitions $(C_p^k)_{p=1}^{P_k}$. Then, for a fixed partition of the functions $(G_k)_{k=1}^{K}$, the problem is to optimize on $(P_k)_{k=1}^{K}$ an additive measure of quality $E((P_k)_{k=1}^{K}) = \sum_{k=1}^{K} R_k(P_k)$ with R_k obtained from equation (3). Let us define $S(l,p)$ has the minimal value of $\sum_{k=1}^{l} R_k(P_k)$ with the constraint $\sum_{k=1}^{l} P_k = p$. The $S(1,p)$ are readily obtained from the optimal segmentation algorithm. Then the additive structure of E shows that $S(l,p)$ is the minimum over u of $S(l-1,u) + R_l(p-u)$, where $R_l(p-u)$ is again obtained via the optimal segmentation algorithm. Given all the $R_l(p-u)$ the calculation of $S(K,P)$ has therefore a cost of $O(KP^2)$.

The final clustering and segmentation algorithm is then obtained by replacing step 2 from the previous algorithm by the following steps:

2.a. for each cluster G_k compute $R_l(p)$ for all $1 \leq p \leq P-K+1$

2.b. find the optimal allocation $(P_k)_{k=1}^{K}$ via dynamic programming

The total cost of those steps is in $O((KP+N)M^2 + KP^2)$. They dominate the cost of each iteration of the K means algorithm provided $M > K$ which seems to be a reasonable assumption.

Even if the polynomial cost of this optimal segmentation is reasonable, it can be quite large compared to the cost of the standard K means applied to functional data considered as high dimensional vectors ($O(MNK)$ per iteration). As the K means algorithm find only a local optimum of its cost function, the common practice is to restart it several times from different random starting points. In order to speed up our algorithm, we propose the following two phases scheme. In a first phase, the standard K means algorithm is applied to the $s_i(t_l)$ considered as vectors from $\mathbb{R}^M$. This is done repeatedly to limit the risk of falling in a local minimum. Then, the best partition obtained this way is used as a starting point for the complete clustering and segmentation algorithm.

4 Experimental results

The proposed method has been tested on the Topex/Poseidon satellite dataset[1]. The Topex/Poseidon radar satellite has been used over the Amazonian basin to produce $N = 472$ waveforms sampled at $M = 70$ points (see, e.g., [3] for details on this dataset). The curves exhibit a quite large variability induced by differences in ground type below the satellite during data acquisition. Figure 1 illustrates the results obtained by our method for $K = 12$ and $P = 60$ (i.e., in average 5 segments per cluster). We have used 20 random starting configurations for the standard K means and selected the best partition as the initial one for the full clustering and segmentation algorithm.

[1] Data are available at `http://www.lsp.ups-tlse.fr/staph/npfda/npfda-datasets.html`

The differences in segmentation strategies between clusters are obvious and show that the algorithm allocates resources as expected. Small clusters and simple ones obtain a simple description while larger or more complex clusters (e.g., with curves with high variability) get more resources for a more accurate representation. The final value of the quantization error as defined by equation (3) is 574×10^3. In the case of a uniform segmentation of each prototype in 5 segments, the error increases slightly to 582×10^3. Moreover, using 5 segments in each cluster fails to show the large differences in complexity of those clusters.

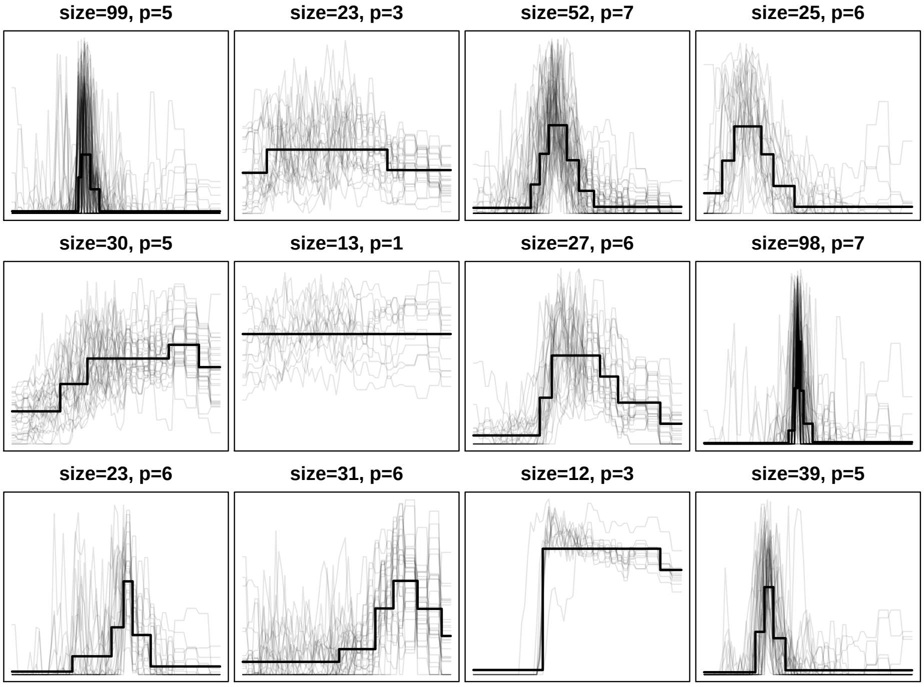

Fig. 1: Results on the Topex/Poseidon dataset with $K = 12$ and $P = 120$

5 Possible extensions

While we have focused here on one particular implementation of the clustering and segmenting idea, it can be implemented in many different ways. It is possible for instance to replace the quadratic criterion $\left(s_i(t_l) - \mu_p^k\right)^2$ by another one, e.g., an absolute difference or a maximal squared error on sub-intervals. As long as the segmentation criterion remain additive, dynamic programming can be used to obtain an optimal prototype efficiently. This can be used to provide a piecewise linear prototype, for instance.

Moreover, a greedy approach (based on hierarchical clustering, see e.g. [5]) can be used to provide efficiently a good but suboptimal segmentation when the criterion is no more additive (e.g., when the prototype is represented by a cubic spline). The resource allocation part is also somehow independent from the segmentation part: as long as the global clustering criterion is additive, optimal allocation can be done efficiently with dynamic programming. This allows to use more complex clustering scheme such as neural gas or self organizing maps in their batch versions.

References

[1] C. Abraham, P.-A. Cornillon, E. Matzner-Lober, and N. Molinari. Unsupervised curve clustering using b-splines. *Scandinavian Journal of Statistics*, 30(3):581–595, September 2003.

[2] R. Bellman. On the approximation of curves by line segments using dynamic programming. *Communication of the ACM*, 4(6):284, 1961.

[3] S. Dabo-Niang, F. Ferraty, and P. Vieu. On the using of modal curves for radar waveforms classification. *Computational Statistics & Data Analysis*, 51(10):4878–4890, June 2007.

[4] B. Hugueney. Adaptive segmentation-based symbolic representations of time series for better modeling and lower bounding distance measures. In *Proceedings of 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2006*, volume 4213 of *Lecture Notes in Computer Science*, pages 545–552, September 2006.

[5] C. Krier, F. Rossi, D. François, and M. Verleysen. A data-driven functional projection approach for the selection of feature ranges in spectra with ica or cluster analysis. *Chemometrics and Intelligent Laboratory Systems*, 91(1):43–53, March 2008.

[6] Y. Lechevallier. Classification automatique optimale sous contrainte d'ordre total. Rapport de recherche 200, IRIA, 1976.

[7] Y. Lechevallier. Recherche d'une partition optimale sous contrainte d'ordre total. Rapport de recherche RR-1247, INRIA, June 1990. `http://www.inria.fr/rrrt/rr-1247.html`.

[8] J. Ramsay and B. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer Verlag, June 1997.

[9] F. Rossi, B. Conan-Guez, and A. El Golli. Clustering functional data with the SOM algorithm. In *Proceedings of XIIth European Symposium on Artificial Neural Networks (ESANN 2004)*, pages 305–312, Bruges (Belgium), April 2004.

[10] F. Rossi, D. François, V. Wertz, and M. Verleysen. Fast selection of spectral variables with b-spline compression. *Chemometrics and Intelligent Laboratory Systems*, 86(2):208–218, April 2007.

[11] F. Rossi and Y. Lechevallier. Constrained variable clustering for functional data representation. In *Proceedings of the first joint meeting of the Société Francophone de Classification and the Classification and Data Analysis Group of the Italian Statistical Society (SFC-CLADAG 2008)*, Caserta, Italy, June 2008.