Functional SOM for variable-length signal windows

A. De Decker, G. de Lannoy, M. Verleysen
Université catholique de Louvain, DICE, 3 pl. du Levant, B-1348 Louvain-la-Neuve,
Belgium
{dedecker, delannoy, verleysen}@dice.ucl.ac.be.

Keywords: SOM, Functional data analysis, signal processing, sampling

Abstract — Functional data are often sampled at high frequency which leads to high-dimensional vectors. The curse of dimensionality makes this type of signal difficult to handle with standard data analysis tools. Functional data analysis uses the functional nature of data to project them on a smooth basis. This paper shows how to extend functional Self-Organizing Maps (SOM) to signal windows having different lengths using functional data analysis. This technique may be applied for example on regularly sampled signals, for which the duration of each signal is varying; an example concerns electrocardiography (ECG), where the signal is usually cut according to the variable period between two heart beats.

1 Introduction

Many real-world data have a high-dimensional nature. For instance, time series sampled at a high frequency, spectrometric data, images, speech and sounds, etc. are usually represented as high-dimensional vectors. Such high-dimensional data are difficult to analyze with traditional machine learning algorithms. Indeed, learning from a few observations in a high-dimensional feature space might result in several difficulties related to the curse of dimensionality (empty space phenomenon, meaningless use of traditional distance measures, etc.) and a high sensibility to noise.

Some of these high-dimensional data have a functional nature, i.e. they origin from a smooth function over a variable, usually the time. When this is the case, functional data analysis (FDA, see [1]) is a common way to overcome the effects of the curse of dimensionality. FDA transforms high-dimensional data vectors in discretized functions. Data analysis is then performed on the projections of the functional representations of the data on a chosen subspace; rather than working on the raw high-dimensional vectors, one can use the coefficients of the projection of each data on the basis for further analysis. Thus, FDA is a fast and easy way to represent accurately high-dimensional vectors by a chosen number of numerical coefficients.

The choice of the subspace basis relies on the prior knowledge of the problem. A Fourier basis allows good representation of periodic data, while a fixed basis handles easily missing values. Another common choice is to use B-splines as basis functions because of their smoothness and projection properties.

Using FDA has another advantage: working on the numerical coefficients of the functional representations of the data leads to a much lighter computational load. Furthermore, functional preprocessing allows fast computation of some specific operations such as integration, derivation, etc.

Another difficulty may be related to the analysis of real-world signals cut in windows whose length differ. Such situation may arise in different contexts; in speech recognition for example, the duration of the same vowel might vary when pronounced several times even by the same speaker. If the sampling rate is kept constant, which is usually the case, it results in vectors with different (high) dimensions. Another example is when one wants to analyze heartbeats from an electrocardiogram (ECG) recording; as the ECG signal is usually cut in windows according to the period between beats, and as the heart rate frequency varies over time, this also results in sampled vectors of different dimensions.

In these cases, it is not possible to directly use classical machine learning algorithms, as the latter usually rely on data represented in a single vector space of fixed dimension. In this paper, we show that this problem can be solved using an appropriate preprocessing. FDA can be used to project functional data observations on subspaces spanned by a fixed number of basis functions, even if the sampled functional data have different dimensions. The idea consists in dilating each of the basis functions proportionally to the length of the raw data vector. In this way, each data vector can be accurately described by a fixed number of numerical coefficients. This paper extends the functional SOM presented in [2] to variable-length signal windows using this methodology. The VLSWF-SOM (Variable-length signal windows functional SOM) is applied to classify heartbeats from an ECG recording.

The following of this paper is organized as follows. Functional data analysis and how it can be used on variable-length signal windows will be explained in section 2. Section 3 will present an application of this methodology on typical ECG data. Section 4 concludes and discusses the results of the application to ECG data.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

2 Functional representation of variable-length signal windows

Variable-length signal windows sampled at fixed frequency results in variable-size vectors. This section shows how a functional preprocessing may overcome this problem, and how to use it for further analysis by SOMs.

2.1 Functional data analysis

The basic idea of FDA is that the discrete vectors acquired when recording smoothly time-varying data can be thought of as functions. (of course the same idea can be developed for functions of another argument than time, for example the wavelength in smooth spectra) In practice, many measured signals such as time series, have this functional property, but they are acquired in a discrete way. FDA provides a simple way to convert discrete functional data into a true functional form. The main FDA assumption is that, for a data vector $y \in \Re^n$, there is a smooth function *f* such that

$$\forall k \in \{1,...,n\}, y_k = f(x_k) + \varepsilon_k ,$$

where $x \in \Re^m (m \geq n)$ is the vector of x_k and $\varepsilon \in \Re^n$ is the vector of errors ε_k. The function *f* can be different for every data vector and is not known. FDA makes it possible to transform the *y* vectors into functions in the $L^2(\Gamma)$ Hilbert space.

However, it is not possible to manipulate arbitrary functions living in a Hilbert space on a computer. A solution to this problem is to work, not on the functions themselves, but on an approximation of each function *f* by projecting it on a chosen subspace; further analysis is then performed on these projections, more specifically on the coefficients of the projection. This technique allows indirect computer handling of the functional data.

The subspace chosen for the projection is a Hilbert basis of the functional space $L^2(\Gamma)$. From this basis, we choose a set of functions $(\Phi_i)_{i \in \aleph}$. B-splines bases are often used; indeed, B-splines have interesting practical properties such as smoothness, numerical stability, locality, efficient calculations, etc.

Then, it is possible to work on by the projection of the functional representations of the data vectors on the vectorial space defined by the first *p* basis functions. The FDA principle necessitates fixing parameter *p* beforehand, according to expert knowledge of the data, or according to some cross-validation scheme. Each observation vector $y \in \Re^n$ is now represented by a vector $\alpha \in \Re^p$ such that

$$\sum_{k=1}^{n} \left(y_k - \sum_{i=1}^{p} \alpha_i \Phi_i(x_k) \right)^2 ,$$

the reconstruction error of the function in the chosen subspace, is minimized.

The FDA approach has nice interesting properties. First of all, it easily allows dealing with irregular sampling. As long as the bases remain fixed, the data observation vectors are changed into a fixed size coordinate vector on the chosen basis. Secondly, FDA provides a simple way to reduce the dimensionality of a problem without loosing too much information. Thirdly, a projection of the observations on a low-dimensional space is an easy way to decrease the noise in the data.

2.2 Variable-length signal windows

It is in general not possible to analyze variable-length vector observations by classical analysis methods. Indeed most conventional data analysis tools, including the SOM and most of its extensions, rely on vector data represented in a single vector space. Therefore, it is interesting to develop an easy and computationally light way to transform variable-length vector observations to a fixed-size representation. When the initial data have a functional nature, this can be done using FDA.

We propose to project the raw data vectors having different lengths on a fixed-sized subspace using FDA. For each functional data to be projected, the basis (for example the set of B-splines) will have the same number dimension *p*, i.e. the same number of basis functions. However, the basis functions themselves will differ between one functional data to be projected and another, in the sense that they will be scaled (according to their *x*-axis, i.e. time) proportionally to the total length of the functional data to be projected. Figures 1 and 2 show examples of similarly shaped signals of different length and the basis functions used for their approximation.

The same number of basis function will be used, but these functions will span over a larger domain if the approximated function if longer. The coefficients of the projection the basis functions scaled according to the data will reflect the form of the functional data, rather than their length, which is the goal to be reached.

After this projection on a scaled version of the basis function, it is possible to use the α_i, i.e. the numerical coefficients of the projections, in classical machine learning analysis methods. This is the idea of the variable-length signal windows functional SOM.

It must be stressed that in addition to a way to reduce in a functional way the dimensionality of the data, this method gets rid of the variable-length observations problem in a straightforward, fast and easy to implement way.

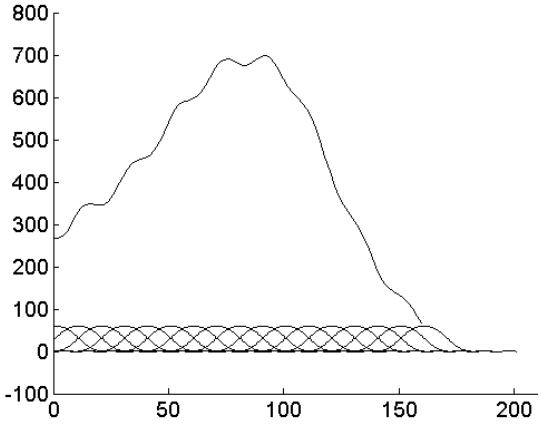

Figure 1 : A function to approximate and its basis functions

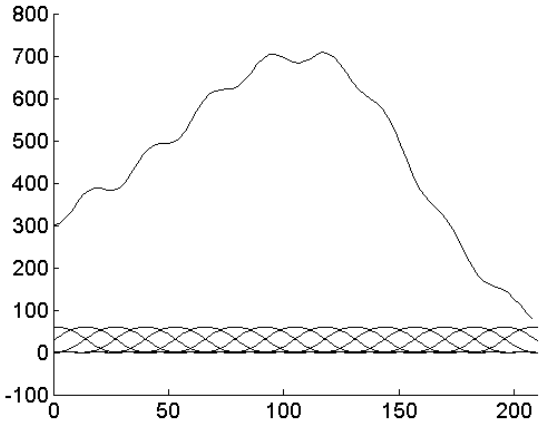

Figure 2 : Another function to approximate and its basis functions

2.3 Functional SOM on variable-length signals

The idea behind the functional SOM [2] is that functions derived from the raw data are used as inputs to a classical SOM [3]. Theoretically, it is possible to use SOMs in a vector space of any finite dimension. However, in general, it is not possible manipulate function in computer calculations. As detailed in Section 2.1, one way to overcome this problem is to submit the numerical coefficients of the functional data projection as inputs for the SOM, rather than the functional data themselves. As these coefficients are the best approximation (in the mean square sense) of the functions according to the chosen basis, the SOM will be trained on low-dimensional vectors which are fairly good approximations of the functional data. This approach is also less computationally intensive than working with the high dimensional raw input vectors. In addition, variable-length signal windows can be dealt with according to the procedure detailed in Section 2.2.

However, SOMs, as most other data analysis tools are based on comparisons in terms of distances between objects (here functions). In order to guarantee the equivalence between the comparison of the functions and their projection on a basis, it is necessary to manipulate the coefficients of the projection if the basis functions are not orthogonal [2]. Defining Φ as the matrix of scalar products between basis functions ($\Phi_{ij} = <\Phi_i, \Phi_j>$), Φ can be written according to its Cholesky decomposition:

$$\Phi = U^T U .$$

Then $U\alpha$ should be submitted to the classical SOM, α being the coordinate vector of each input function. In this way, the theoretical functional SOM can be efficiently approximated by a SOM on vector data.

3 Application to ECG data

A typical example of variable-length signal windows concerns the analysis of ECG signals. Indeed such signals are usually cut into windows whose lengths correspond to the period between two beats (or a shifted version of it). As the heart beat frequency varies over time, this results in variable-length windows.

3.1 ECG data and problem description

An electrocardiogram is a marker of the heart activity. The heart has four cavities: two atria and two ventricles. When those muscles are activated, their movement is accompanied by an electrical depolarization. The ECG records the electrical signals the heart activity produces. At the beginning of a heartbeat, the atria contracts and produces the P wave visible on Figure 3. Next, the ventricles beat, producing the QRS complex. The last part of a heartbeat is produced by the repolarisation of the ventricles, which produces the T wave. This cycle forms one heartbeat. The heart of a healthy person beats between 60 and 100 times per minute.

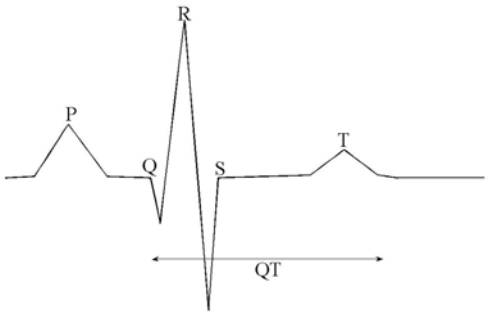

Figure 3 : ECG beat and its different peaks

An ECG recording is a powerful tool which can help cardiologists to detect pathologies. The length (time)

between different parts of the beat (PR, QRS, QT (see Figure 3), etc) can be used to detect several types of heart malfunctions. However, a specialist is needed for correct interpretation of an ECG recording. In the past, ECG recording used to be only a few seconds long, allowing a manual handling and inspection by cardiologists. Nowadays, it is more and more frequent to use much longer ECG recordings. Sometimes, the heart activity is recorded for 24 hours. Such ECGs hold much more information, but it is not possible anymore to analyze them manually. In addition, some analyses might only be done on good quality beats: flat baseline, good signal to noise ratio, etc. It is not possible for a cardiologist to scan manually the whole recording in order to find portions of good quality signals usable for the analysis. This is why there is a need for an automatic classifier able to select beats suitable for fine analysis (which will be called "good beats" in the following) from beats that are not suitable (which will be called "bad beats") in a long ECG recording.

In this experiment, we used a 15-minute ECG recording sampled at 500 Hz recorded on healthy patients. The aim of this work is to discriminate good quality beats (see Figure 4) from bad quality ones (see Figure 5). Each one of those beats having different lengths.

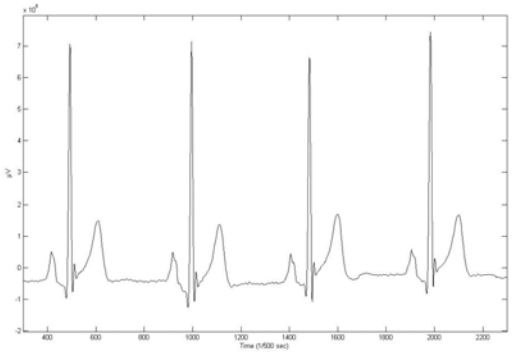

Figure 4 : good quality beats

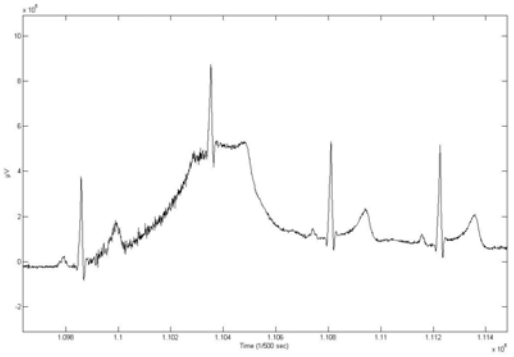

Figure 5 : bad quality beats

3.2 Adaptation of the method to ECG data

In order to apply FDA on the different beats, the first thing is to cut the 15 min long signal in single beats. The signal was cut at the R peaks, so that one beat starts at the maximum of the QRS complex and ends and the maximum of the next QRS. Cutting the signal at the R peaks is a far from being an ideal solution. It cuts the beats at the middle of their natural structure. However, the R detection is much more robust than the detection of the other peaks. In addition, and because we want to classify the beats by their shapes, it is important to be sure that their most intensive peak is perfectly aligned to avoid errors due to bad peaks alignment.

To detect the R peak, a simple local maximum detection was used. The maximum in a window of fixed length was annotated as the maximum of the QRS complex. The window was shifted forward by 200 ms after the detection of an R peak (a heartbeat cannot occur physiologically less than 200 ms after the previous beat, [4]). As the ECG is sampled at a constant rate and the heart frequency changes with time, the beats cut from the original ECG signal are of unequal length. A total of 1017 windows results from this procedure applied to the 15-minutes ECG recording.

Next, FDA was applied in order to bring back the different beats at the same length. A B-splines basis with 60 basis functions of order 4 was used. The number of B-splines was chosen using a leave-one-out method as in [5] in order to minimize the reconstruction error of the signals. Figure 6 shows the reconstruction error versus the number of B-splines in the basis.

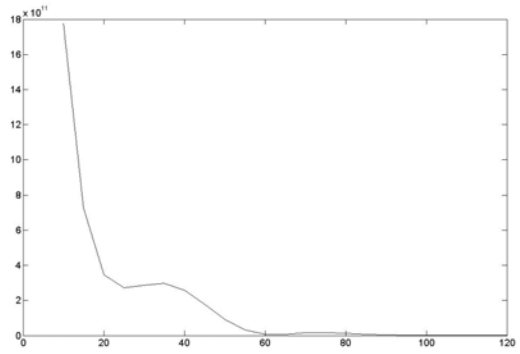

Figure 6 : Reconstruction error in function of the number of B-splines

The reconstruction error clearly decreases up to 60 B-splines. A higher number of B-splines does not result in significantly improved reconstruction. For this reason, 60 B-splines were used to approximate each beat, as this offered a good compromise between reconstruction error and dimensionality reduction.

The next step was to apply functional SOM to the coefficients of the splines approximation of each of the

1017 beats. A large SOM with 15x15 prototypes was trained using the SOM toolbox 2.0[1]. The SOM prototypes were initialized along the highest eigenvectors of the input vectors. The training consisted first on a sequential training of 10 epochs, with the initial and final training radii respectively set to 6 and 1, and a Gaussian neighborhood function; this part of the training is essentially responsible for the topographic organization property of the SOM. Next, a batch training of 500 epochs was performed, with a training radius set to 0, in order to guarantee the fine-tuning of the vector quantization property of the map.

After the training, a k-means clustering was applied on the prototypes of the SOM. The number of clusters was set to 3 after a visual analysis of the prototypes on the map (see Section 3.3 for details about this choice). The goal is to classify the beats into "good" and "bad", leaving opportunities for the "bad beats" to be classified in several (here 2) clusters.

3.3 Results and discussion

The application of functional SOM to the ECG data described in section 3.2 shows a nicely structured map with three very clear clusters (see Figure 7). The upper left corner of the map groups beats with a descending baseline while the lower right corner of the map has beats with ascending baselines. The good beats which should be selected for the analysis are grouped along the diagonal going from the lower right corner to the upper left corner. The very clear organization of this map suggested using 3 clusters for the k-means algorithm on the prototypes.

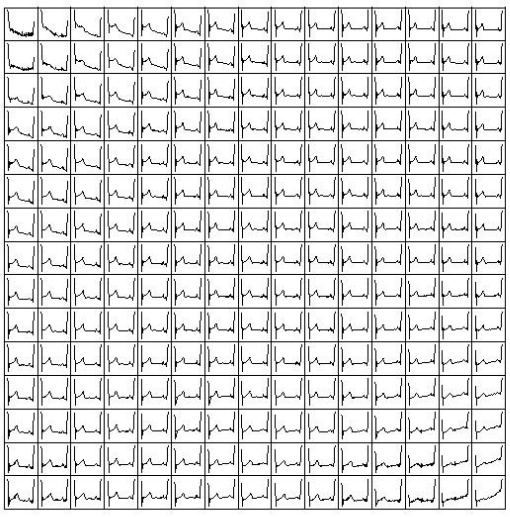

Figure 7 : SOM prototypes

[1] Available from http://www.cis.hut.fi/projects/somtoolbox/

Figure 8 shows the results of the k-means clustering with 3 clusters on the map showed in Figure 6. The clustering confirms our visual analysis on Figure 6. Cluster 1 contains the beats with a descending baseline which are on the upper left corner of the map. On the diagonal, cluster 2 contains the good beats. Cluster 3 groups the ascending baseline beats from the lower right corner of the map.

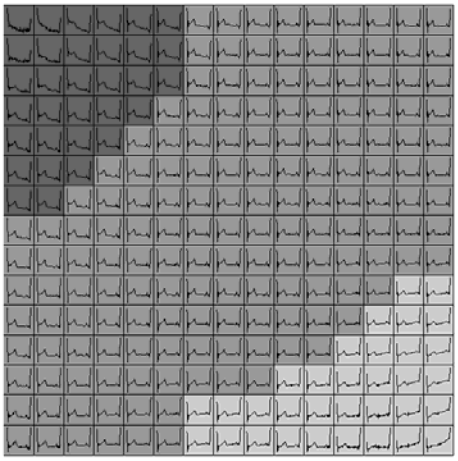

Figure 8 : Clustered SOM prototypes

71.39% of the beats are classified into cluster 2 (“good beats”) while cluster 1 and 3 represent respectively 14.26% and 14.36% of the whole recording.

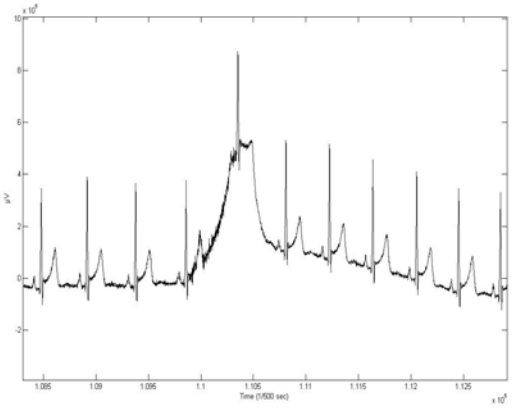

Figure 9 : non classified bad beats

This result seems acceptable as the overall baseline of the recording is constant, and the clusters 1 and 3 have opposite trends. Figure 9 shows a part of the ECG recording containing beats that should be rejected (according to visual inspection) for further analysis. Figure 10 shows the same part of the recording colored with respect to the clustering. The classification seems to be accurate enough to discriminate between good and bad beats.

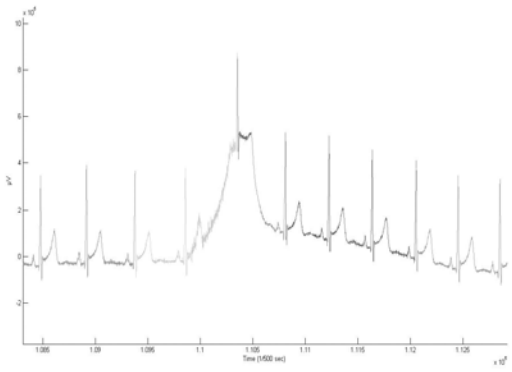

Figure 10 : Classification of bad beats

If the method behaves correctly, beats of different lengths, but with similar shapes should be classified in the same cluster. Figure 11 shows an example of the results obtained from this type of beats. The beats in the rectangle are slightly shorter than the other normal beats; still, their shape is similar to them. The classification method puts all those beats in the second cluster, which corresponds to the good beats.

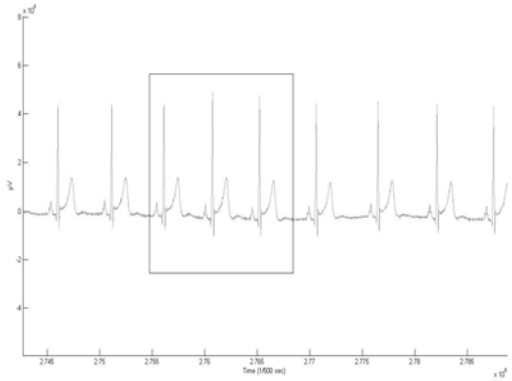

Figure 11 : Classification of good beats

4 Conclusions

This paper shows that it is possible to use classical machine learning methods on variable-length observation data using FDA preprocessing. Adapting the size of the basis functions used in the preprocessing allows getting rid of differences between the observation lengths, and concentrating on their shape. The method has been applied to discriminate between normal shaped heartbeats and abnormal ones, using a functional SOM approach and further k-means clustering to group the cluster prototypes into a low number of classes.

Further work will consist in extending the method to local time variations in signal windows, adapting the widths of basis functions individually through gradient descent on a local reconstruction error.

Acknowledgements

Part of this work has been funded by the Belgian "Région de Bruxelles-Capitale", Beats project. The experimental data and ECG expertise have been provided by Cardionics s.a.; the authors would like to thank M. Jean Waldura and Cardionics team for discussions and support.

References

[1] Jim Ramsay and Bernard Silverman. Functional Data Analysis. Springer Series in Statistics. Springer Verlag, June 1997.

[2] Fabrice Rossi, Brieuc Conan-Guez, and François Fleuret, “Clustering functional data with the SOM algorithm”, In Proceedings of ESANN 2004, pages 305-312, Bruges, Belgium, April 2004.

[3] Teuvo Kohonen, Self-Organizing maps, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 3rd edition, 2001.

[4] Ivaylo I Christov “Real time electrocardiogram QRS detection using combined adaptive threshold” *BioMedical Engineering OnLine* 2004, 3:28 doi:10.1186/1475-925X-3-28

[5] Fabrice Rossi, Nicolas Delannay, Brieuc Conan-Guez, and Michel Verleysen. “Representation of functional data in neural networks”. *Neurocomputing*, Vol 64, pp 183-210.