Supervised classification of curves via a combined use of functional data analysis and random forest

Fabrizio Maturo and Rosanna Verde

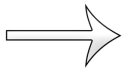
Department of Mathematics and Physics, University of Campania "*Luigi Vanvitelli*", Caserta, Italy

Technology and Environment Workshop 2021, France

The basic idea of this talk

Supervised learning
from high-dimensional data
is a challenging task

Some classical problems:
-Curse of Dimensionality
-Dimensionality Reduction
-Tradeoff between Complexity and Accuracy
-Feature Selection
-Interpretation (black box vs glass box models)



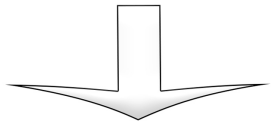
Our proposal is to combine Functional Data Analysis and some Machine Learning techniques

Functional Data Analysis (FDA)

Tree-based supervised classification methods
-Decision Trees (DTs)
-Bagging
-Random Forest (RF)
-Boosting

Dimensionality Reduction

Additional Functional Features



-No curse of Dimensionality
-High Accuracy
-Interpretability
-Original Feature Selection
-Additional Features

Supervised and unsupervised classification

In recent decades, technological advancement led to the development of tools to collect **vast amounts of data** usually recorded at temporal stamps or arriving over time, like data coming from sensors.

When we are dealing with hundreds / thousands of observations for each unit, we need to find techniques to **reduce dimensionality**.

Possibly, these techniques should **help the interpretations** of phenomena rather than complicate them!

A common way of analyzing data also involves **classification** techniques (Aggarwal et al., 2004; Ailon et al., 2009; Gama, 2010; Silva et al., 2012).

Many applications in different areas

Hence, **dimensionality reduction and classification techniques** have assumed an increasingly important role to deal with this type of data.

Indeed, supervised and unsupervised classification have important **applications** in **many areas** such as:

- multimedia processing;
- environmental monitoring;
- industrial quality control;
- speech processing;
- robotics;
- bioinformatics;
- medicine;
- ... and many other fields!

Classification from high-dimensional data is a challenge

Despite constant improvements in the literature, both unsupervised and supervised learning from high-dimensional data are **challenging tasks due to many issues** such as:

- the curse of dimensionality;
- clusters robustness over time;
- inefficiency of most traditional algorithms;
- computational time-consuming;
- the trade-off between complexity and accuracy;
- the trade-off between interpretability and accuracy (black box vs glass box models).

In this talk, we focus on **supervised learning from high-dimensional data**.

Major Limitations

There are many problems using the classical techniques when:

- as the **number of temporal observations increases**, the problem of the curse of dimensionality increases;
- the sampling units are observed in a **finite set of time points** that may be **irregularly spaced** and **different for the same individuals**;
- **algorithm convergence** may be difficult due to possible local minimum and **high dimensionality**;
- we would deal with more recent machine learning techniques, such as tree-based classifiers like bagging, random forest, and boosting, usually their **interpretation is very difficult**.

To solve these issues, the functional data analysis (**FDA**) approach (Ramsay and Silverman, 2005) combined with tree-based classifiers seems to be appropriate.

Functional Data Analysis (FDA)

One of the most recent approaches to **dealing with high-dimensional data** is Functional Data Analysis (FDA). Particularly, when we deal with time series with a huge number of observations.

Of course, FDA can also be used even when the **reference domain is different from time** but, in this context, we focus on the case in which the domain is the temporal one.

The basic idea of FDA is to **deal directly with the function generating the data** instead of the sequence of observations, and thus to treat observed data functions **as single entities** (Ramsay and Silverman, 2005).

Some Advantages of FDA

- Ramsay and Dalzell (1991) stressed that sometimes the aim of a study can be **functional in nature**, and some modeling problems are more natural to consider functionally;
- FDA allows us assessing **important additional sources of pattern and variation**;
- Cuevas (2014) remarked that, differently from time series analyses, **we do not need that data are sampled at equally spaced time points, and no assumptions of stationarity** are necessary;
- Ferraty and Vieu (2006) highlighted that often **crucial information is included in derivatives** rather than in the data themselves;
- There is the theoretical possibility of **observing the phenomenon in a much finer grid** and, in the limit, to observe x(t) at **any fixed instant**.

B-Spline Representation

The first step in FDA is to convert the observed values $z_{i1}, z_{i2}, ..., z_{iT}$ for each unit $i = 1, 2, ..., N$ to a functional form.

The most common approach to approximate functions using a finite representation in a fixed basis (Ramsay and Silverman, 2005), e.g. b-splines.

B-Spline Representation

$$x_i(t) = \sum_{j \in \mathbb{N}} c_j \phi_j(t) \approx \sum_{j=1}^{K} c_j \phi_j(t) = \hat{x}_i(t) \quad (1)$$

where:

- $\hat{x}_i(t)$ is the approximated function for the i-th unit;
- $\phi_j(t)$ are linearly independent and known functions, called basis functions;
- c_j is the vector of coefficients of the linear combination;
- K is the total number of basis used to represent the functions.

Functional Distance

Focusing on the case of an Hilbert space with a metric $d(\cdot,\cdot)$ associated with a norm so that $d(x_1(t), x_2(t)) = \|x_1(t) - x_2(t)\|$, and where the norm $\|\cdot\|$ is associated with an inner product $\langle\cdot,\cdot\rangle$ so that $\|x(t)\| = \langle x(t), x(t)\rangle^{1/2}$, we can obtain as specific case the space $L_2[a,b]$ of real square-integrable functions defined on $[a,b]$ by $\langle x_1(t), x_2(t)\rangle = \int_a^b x_1(t)x_2(t)dt$.

Hence, focusing on the L_2-norm, a **commonly used distance between functional elements** is given by

L_2-distance

$$\|x_1(t) - x_2(t)\|_2 = \left\{ \frac{1}{\int_a^b w(t)dt} \int_a^b |x_1(t) - x_2(t)|^2 w(t)dt \right\}^{1/2} \quad (2)$$

where w are the weight and the observed points on each curve are equally spaced (Febrero-Bande and de la Fuente, 2012).

Semi-metric based on Derivatives

However, many scholars (e.g. Ferraty and Vieu 2006, Febrero and de la Fuente 2012) believe that **Equation 2 does not necessarily provide the more informative proximity measures between functional elements**, and therefore considering other distances between curves could give better information.

For this reason, several metric and semi-metric have been proposed in the literature. Particularly, because of their high informative power, **semimetric proximity measure based on derivatives** are widely adopted.

The **distance between the r-order derivatives** of two curves $x_1(t)$ and $x_2(t)$ can be expressed as follows

Semi-metric of Derivatives

$$d_2^{(r)}\left(x_1(t), x_2(t)\right) = \left[\frac{1}{T}\int_T \left(x_1^{(r)}(t) - x_2^{(r)}(t)\right)^2 dt\right]^{\frac{1}{2}} \tag{3}$$

where $x_1^{(r)}(t)$ and $x_2^{(r)}(t)$ are the r-derivatives of $x_1(t)$ and $x_2(t)$, respectively.

Data reduction via FPCA

Due to the characteristics of functional data, a **reduction dimension technique** is necessary for explaining the main features of the data.

To solve this problem, we can adopt the **functional principal component analysis** (FPCA), which allows us to reduce the infinite dimension of a functional observation by a reduced set of uncorrelated variables.

It allows us displaying the functions by a **linear combination of a small number of functional principal components** (FPCs). The functional data can be rewritten as a decomposition in an orthonormal basis by maximizing the variance of $x(t)$:

Functional Principal Components Decomposition

$$\hat{x}_i(t) = \sum_{i=1}^{K} \nu_{ik}\xi_k(t) \tag{4}$$

where $\nu_{i,k}$ is the score of the generic FPC ξ_k for the generic function x_i $(i = 1, 2, ..., N)$.

Semi-metric based on FPCs

Another widely used **semimetric proximity measures between curves is that based on functional principal components** (e.g. (Cuevas, 2014)).

The basic idea is to exploit the functional principal components decomposition for computing the distance between functional elements as follows:

Semi-metric based on FPCs

$$d_2\left(x_1(t), x_2(t)\right) \approx \left[\sum_{p=1}^{P} \left(\nu_{p,a} - \nu_{p,b}\right)^2 \|\xi_p\|\right]^{\frac{1}{2}} \tag{5}$$

Supervised classification of functional data

In functional classification, the aim is to **predict the class or label** Y of an observation X taking values in a separable metric space (F, d).

For simplicity we will assume that the only possible values of Y are 0 or 1.

Classification of a new observation x from X is carried out by constructing a mapping $f : F \longrightarrow \{0, 1\}$, called a **classifer**, which maps x into its predicted label and whose probability of error is given by $P\{f(X) \neq Y\}$.

In other words, focusing on the **case of a single functional variable**:

- we start from a **training set** of dimension $n \times t + 1$ with which we can reconstruct the curve using T points of the domain for each individual and we also have the information on the label of individuals;
- we build a **classification rule** to assign new statistical units, also represented by curves of which we do not know the label of the group.

Combining functional data and tree-based methods

In recent years, in the literature, functional supervised classification has became a very **attractive and lively** topic.

Some approaches have been proposed, e.g. Logistic Classifier, k-Nearest Neighbor Classifier, Maximum Depth Classifier, and Kernel Classifier.

However, **research on this topic is still in progress**.

We propose to **combine FDA and tree-based classifiers** with **many possible approaches**:

- Using the FPCs decomposition;
- Using b-splines;
- Using wavelets.

Functional Decision Trees (FDTs)

The first step is to **extend Decision Trees** (DTs) to the context of Functional Data (FD).

We will indicate this approach as "**Functional Decision Trees (FDTs)**".

In this talk, **we will focus only on the FPCs** decomposition but keeping in mind that this approach can be **easily be extended to b-splines and wavelets**.

The basic idea is to **exploit the FPCs** decomposition to **build a DT** in order to get an **interpretable classification rule** to assign curves to different groups.

Functional Decision Trees

Exploiting the coefficients of FPCs, DT can be extended to the case of FD of the form $\{y_i, x_i(t)\}$, with a predictor curve $x_i(t)$, $t \in J$, and y_i being the (scalar) response value observed at sample $i = 1, ..., n$.

The response variable could be either numeric or categorical, leading to **regression or classification trees**, respectively; however, here we focus on the case of a binary dependent variable and thus we concentrate on functional classification trees, particularly on the **scalar-on-function classification** problem.

Classification trees consist in **recursive binary partitions of the feature space into rectangular regions (terminal nodes or leaves)**. To build the tree, an **optimal binary partition** is provided at each step of the algorithm, based on the **optimization of cost criterion** (e.g. to decrease the impurity).

The **algorithm** begins with the **full data set composed of the coefficients** obtained with the FPCs decomposition and continues until the terminal leaves are obtained.

Interpretation of Functional Decision Trees

The **rule is replicated** to perform the most suitable binary separation on all resulting nodes. Typically, a huge tree is produced at the beginning, which is then **pruned** according to an optimization criterion.

Therefore, the **coefficients of the linear combination** are used as **new features** to predict the response.

The **interpretation is slightly different** with respect to the classical DT because the values of the splits of ν_{ik} should be interpreted **according to the part of the domain that the single FPC mostly represents**.

Hence, the **joint read of the coefficients and the plot** of $\xi_k(t)$ can help interpreting the classification tree.

Functional Bagging (FB)

The great problem with a single FDT is that its **predictive performance is usually not persuasive**, and modest changes in the data may lead to very diverse FDTs. A useful technique to **reduce this variance** is to create an **ensemble of FDTs using Functional Bagging (FB) (i.e. Functional Bootstrap aggregation)**.

We generate B **different bootstrapped training data sets**. We **train our classifier on the b-th bootstrapped training se**t to get $\hat{f}^{*b}(x(t))$.

We **average all the predictions** to obtain

Avarage of predictions in Functional Bagging

$$\hat{f}_{bag}(x(t)) = \frac{1}{B}\sum_{b=1}^{B} \hat{f_{bag}}^{*b}(x(t)) \tag{6}$$

We record the class predicted by each of the B trees, and take a **majority vote**: the **overall prediction for each new curve is the most commonly occurring class among the B predictions**.

Out-of-Bag Curves (OOBCs) for the error estimation

Because **FDTs are repeatedly fit to bootstrapped subsets of the FD**, one can show that on average, **each bagged FDTs makes use of around 2/3 of the functions**.

We will name the **remaining 1/3 curves** that on average are not used to fit a given bagged FDT as the **out-of-bag curves (OOBCs).**

We can **predict the response y for the i-th curve** using each of the FDT in which that curve was OOB.

This will yield around **B/3 predictions for the i-th curve**, which we **average**. This estimate is essentially the LOO cross-validation error for bagging, if B is large.

Functional Random Forest (FRF)

One **limit of Bagging** is that the **FDTs are correlated** and thus the desirable reduction of variance is not so good.

Indeed, in FB, almost all the **FDTs will be similar** because the upper parts of them will present the **first binary split according to the same FPC** in each FDT.

Keeping in mind that given a set of K independent observations $Z_1, ..., Z_K$, each with variance σ^2, the **variance of the mean** $\bar{Z}$ of the observations is given by $\frac{\sigma^2}{K}$

Functional Random Forests (FRF) provides an improvement over FB by way of a small tweak that **decorrelates the trees**. This **reduces the variance when we average the FDTs**.

Functional Random Forest

The procedure is the same of FB but when building the FDTs, **each time a split in a FDT is considered, a random selection of m FPCs is chosen as split candidates** from the full set of the K FPCs.

The split is allowed to use only one of those m FPCs.

A fresh selection of m FPCs is taken at each split, and typically we choose $m \approx \sqrt{K}$, that is, the number of FPCs considered at each split is approximately equal to the square root of the total number of FPCs considered.

Indeed, on average, $\frac{m-K}{m}$ of the splits **will not even contemplate some FPCs coefficient**. In this way, **FRF decorrelates the FDTs**, making the average of the FDTs less variable and hence more reliable.

If $m < K$, we have FRF.
If $m = K$, we have FB $=$ FRF.

Application to a real dataset derived from twelve monthly electrical power demand time series in Italy

We have a sample of 67 signals. The black signals are the days from Oct to March whereas the red curves are those from April to September.

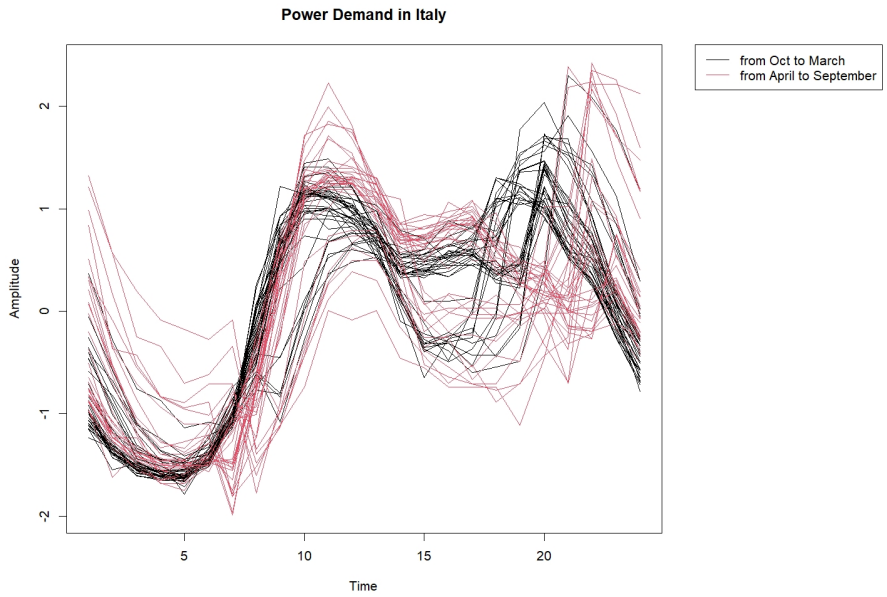


Figure: Power Demand in Italy.

The basic idea is therefore to predict, based on the trend of the curves, whether a new curve belongs to the October-March class or the April-September class.

Smoothed functions of electrical power demand

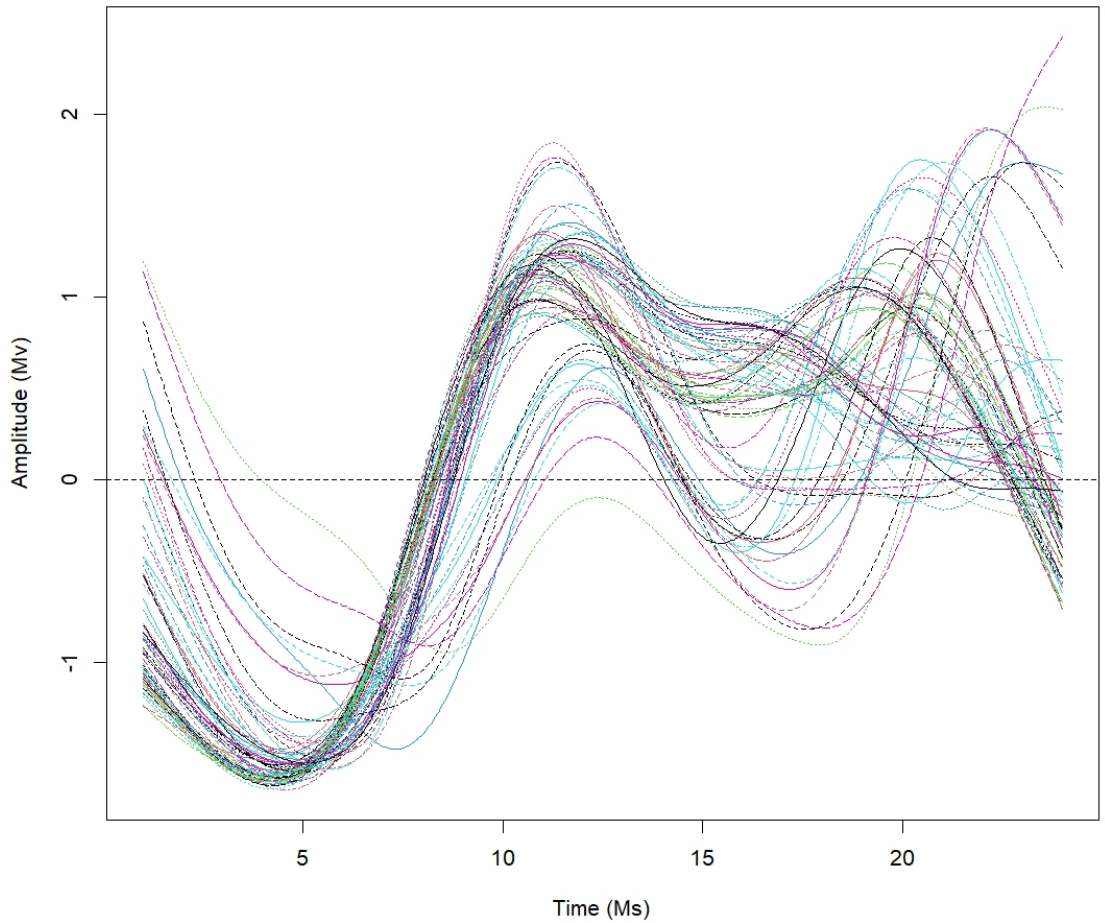


Figure: Smoothed functions using twenty-eight b-splines.

Functional Principal Components Decomposition

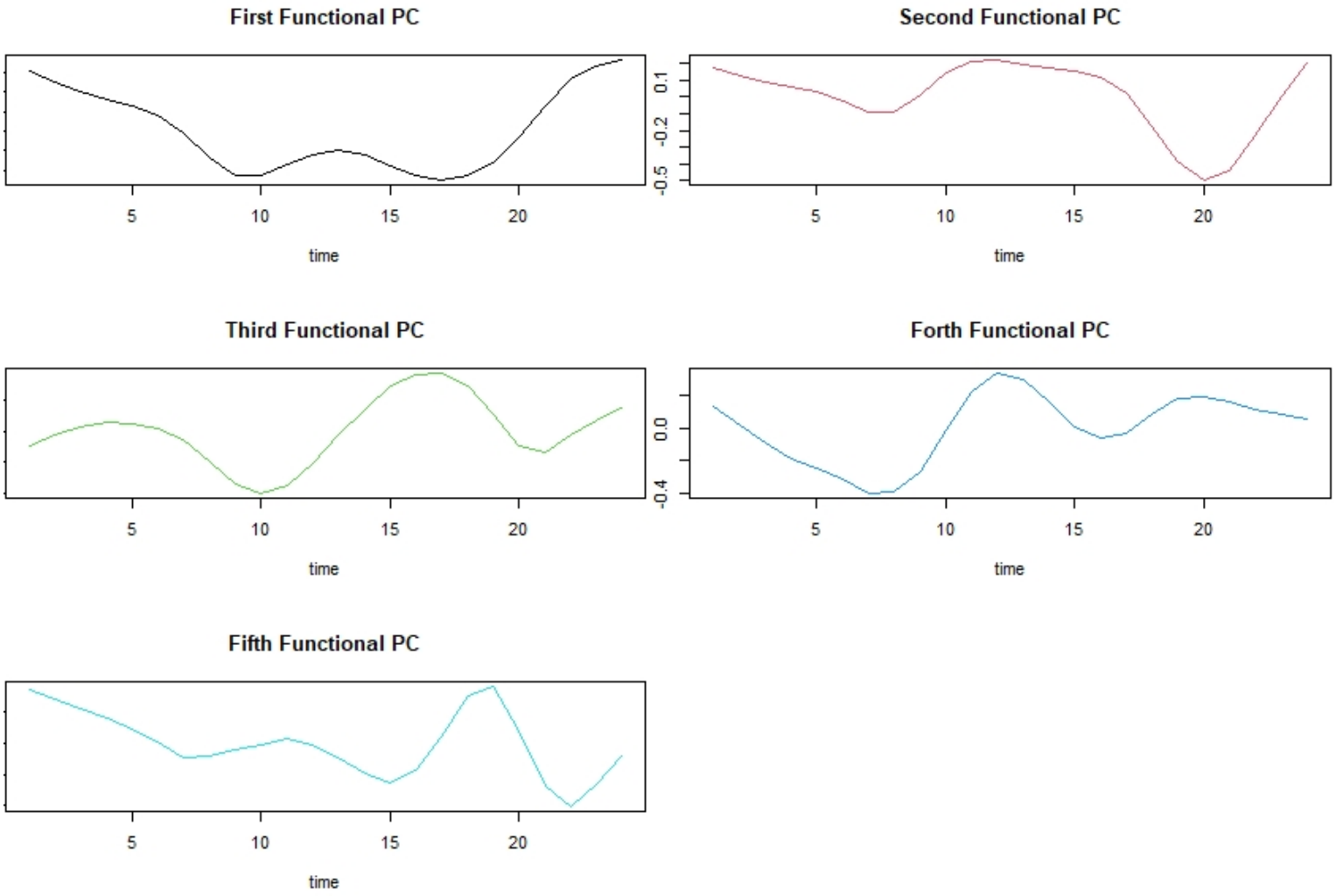


Figure: Functional Principal Components.

Explained Variability of each FPC

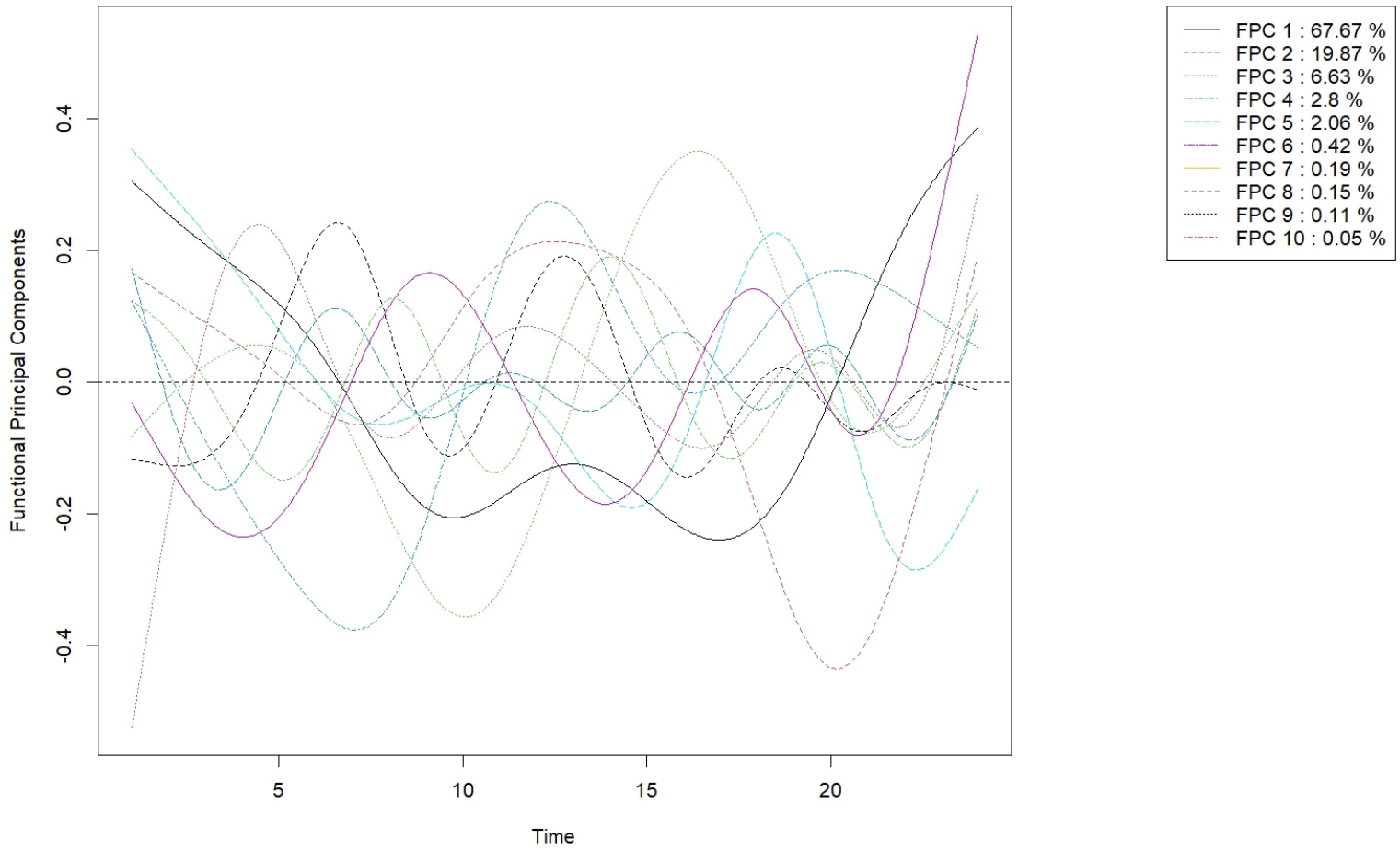
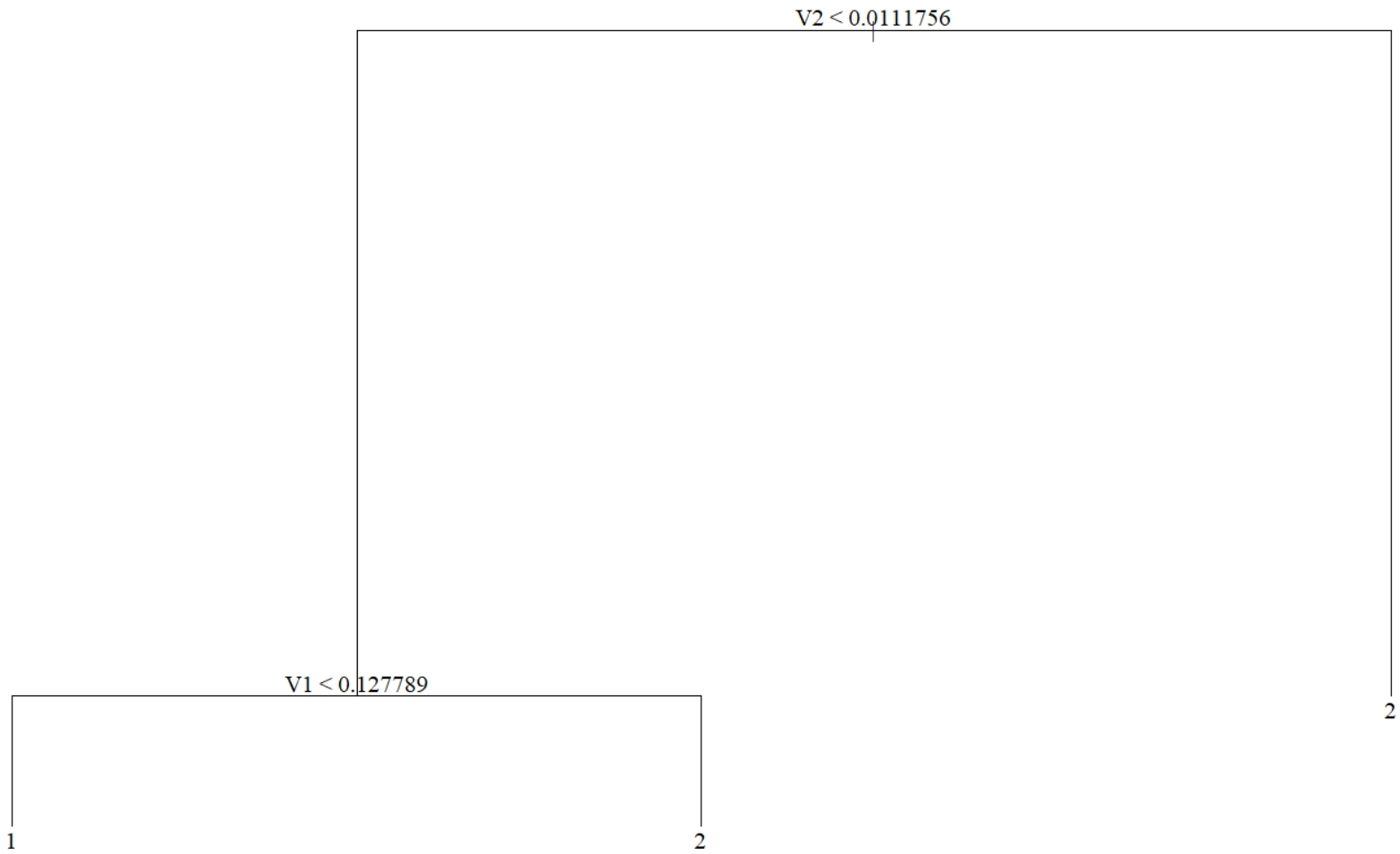


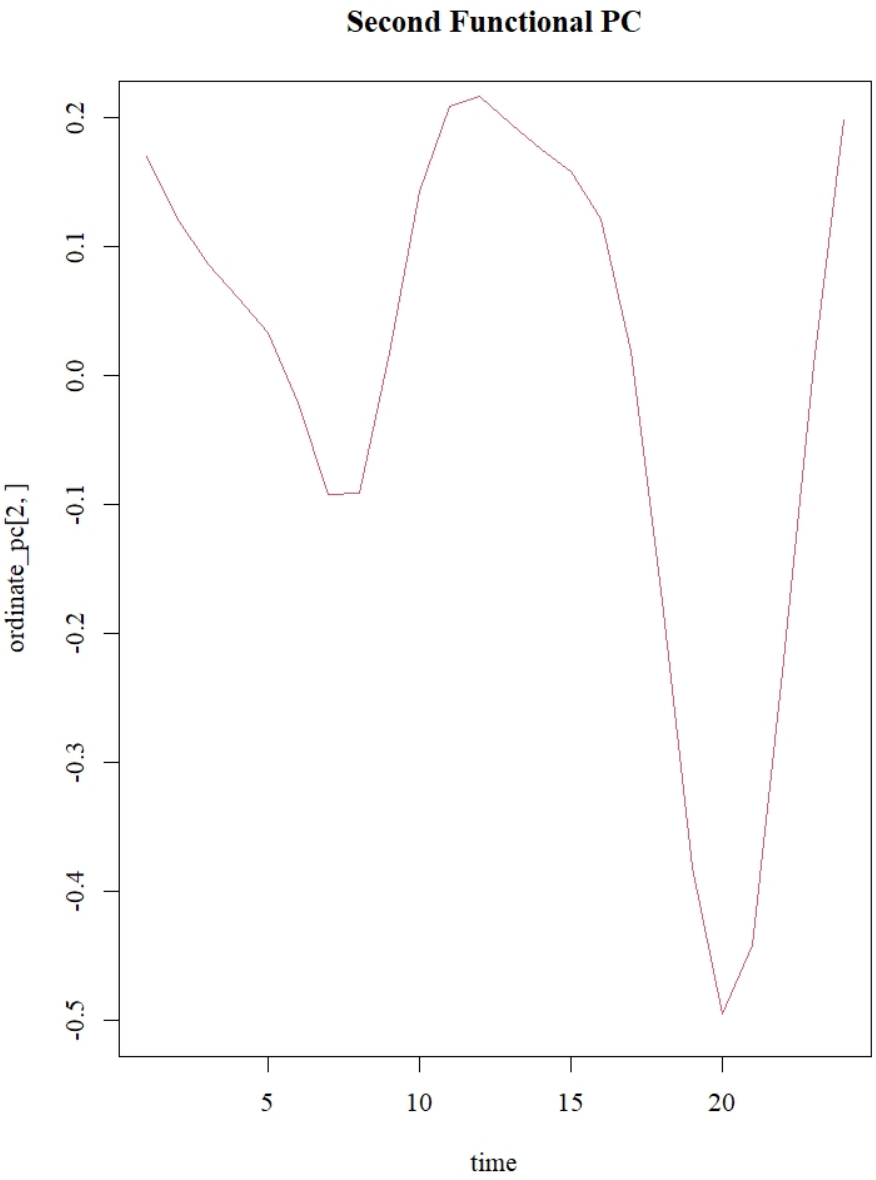
Figure: Functional Principal Components.

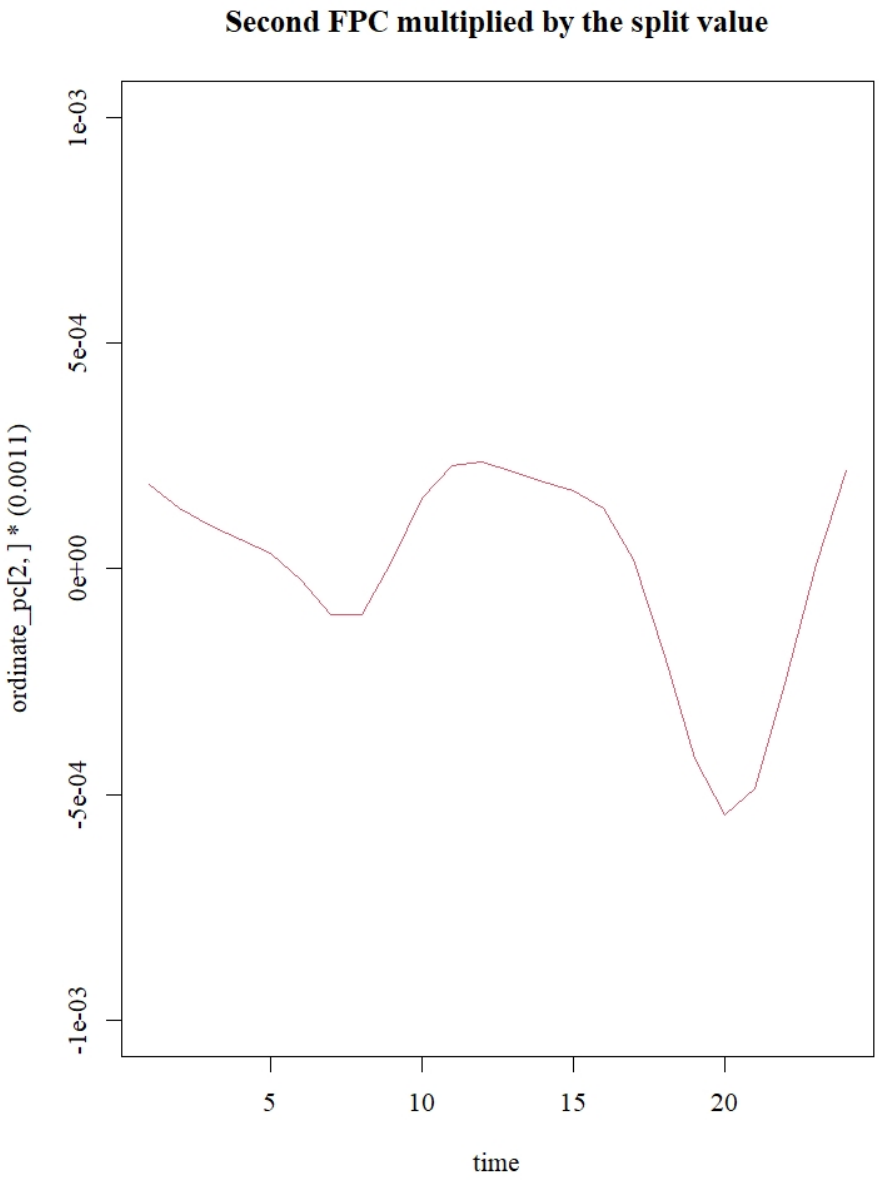
Functional classification tree using FPCs

Misclassification error rate: $0.02985 = 2 \ / \ 67$



Split value interpretation





Original signals vs predicted

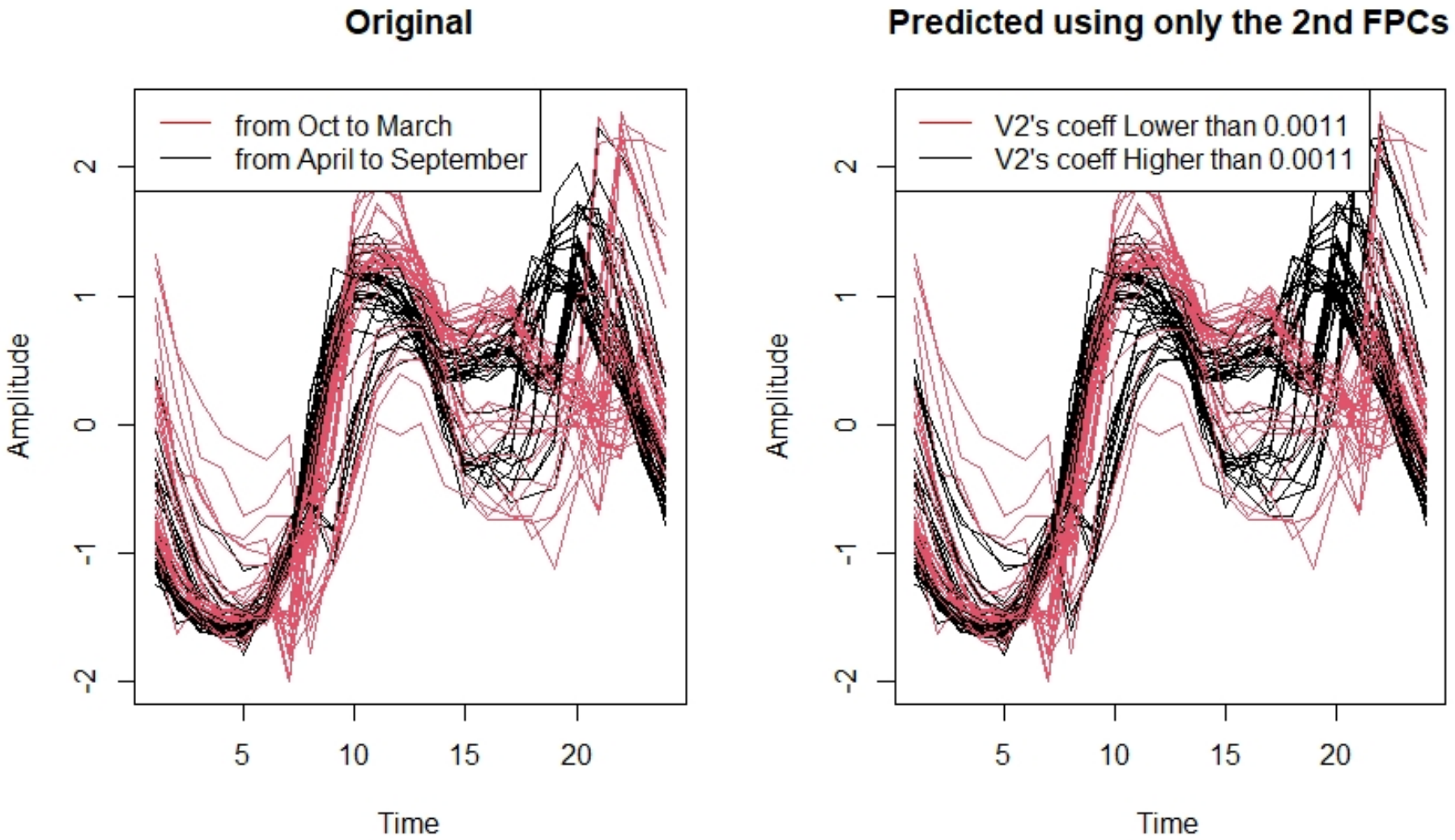
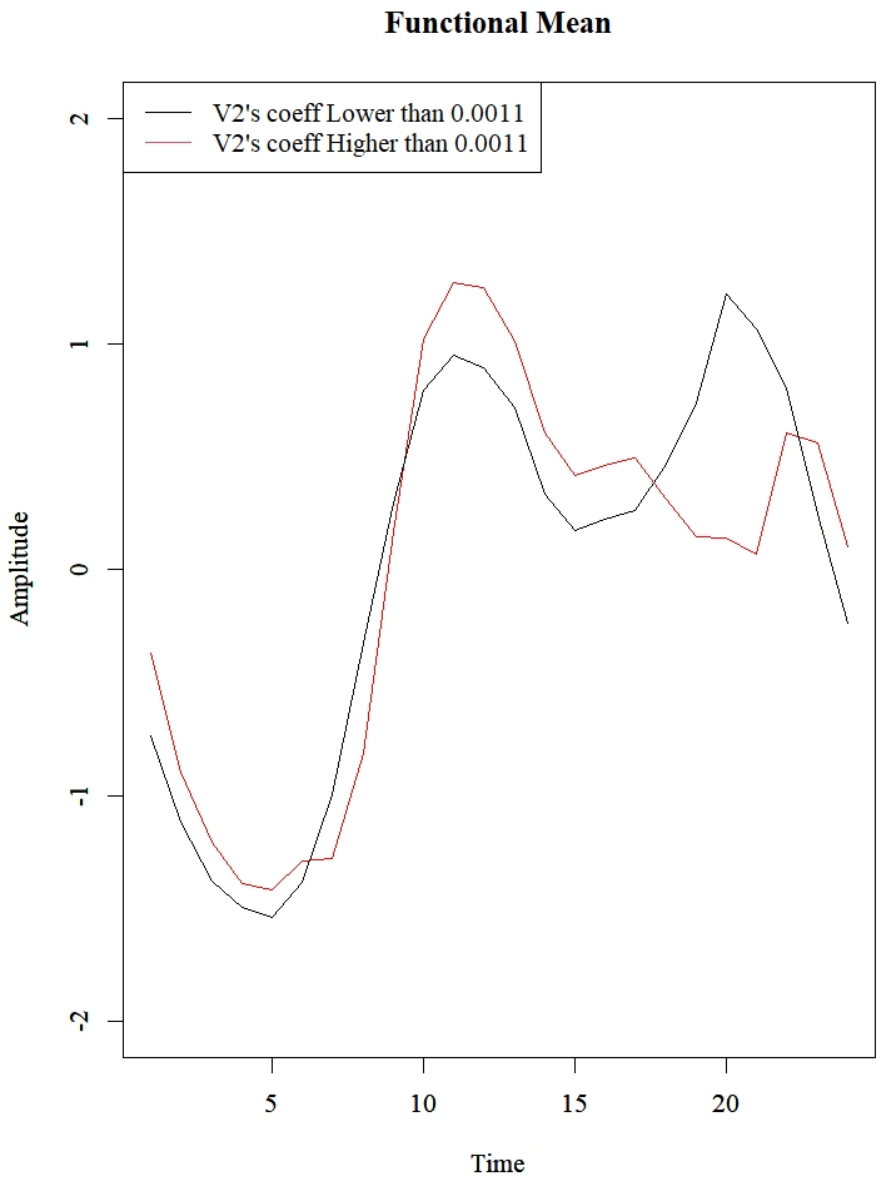
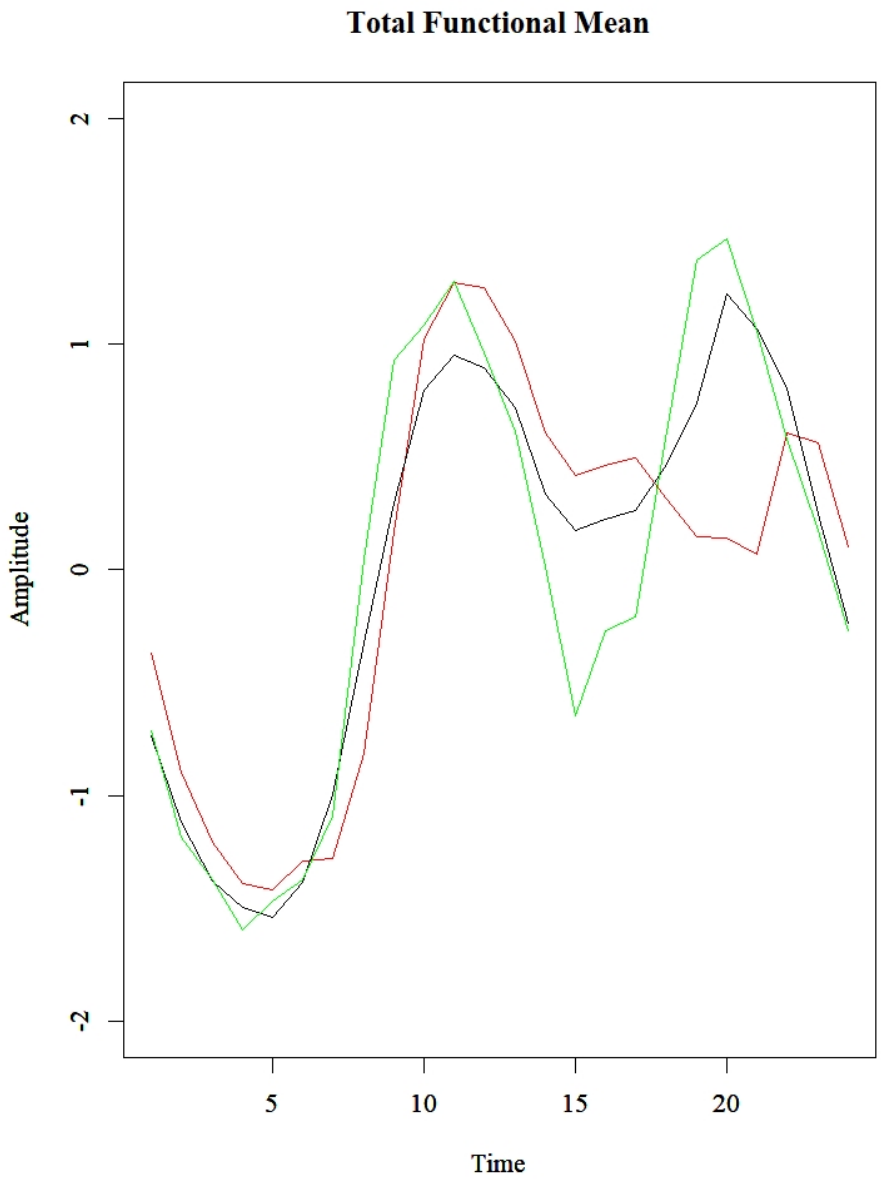


Figure: Original signals vs predicted using only the second FPC with the split at 0.0011

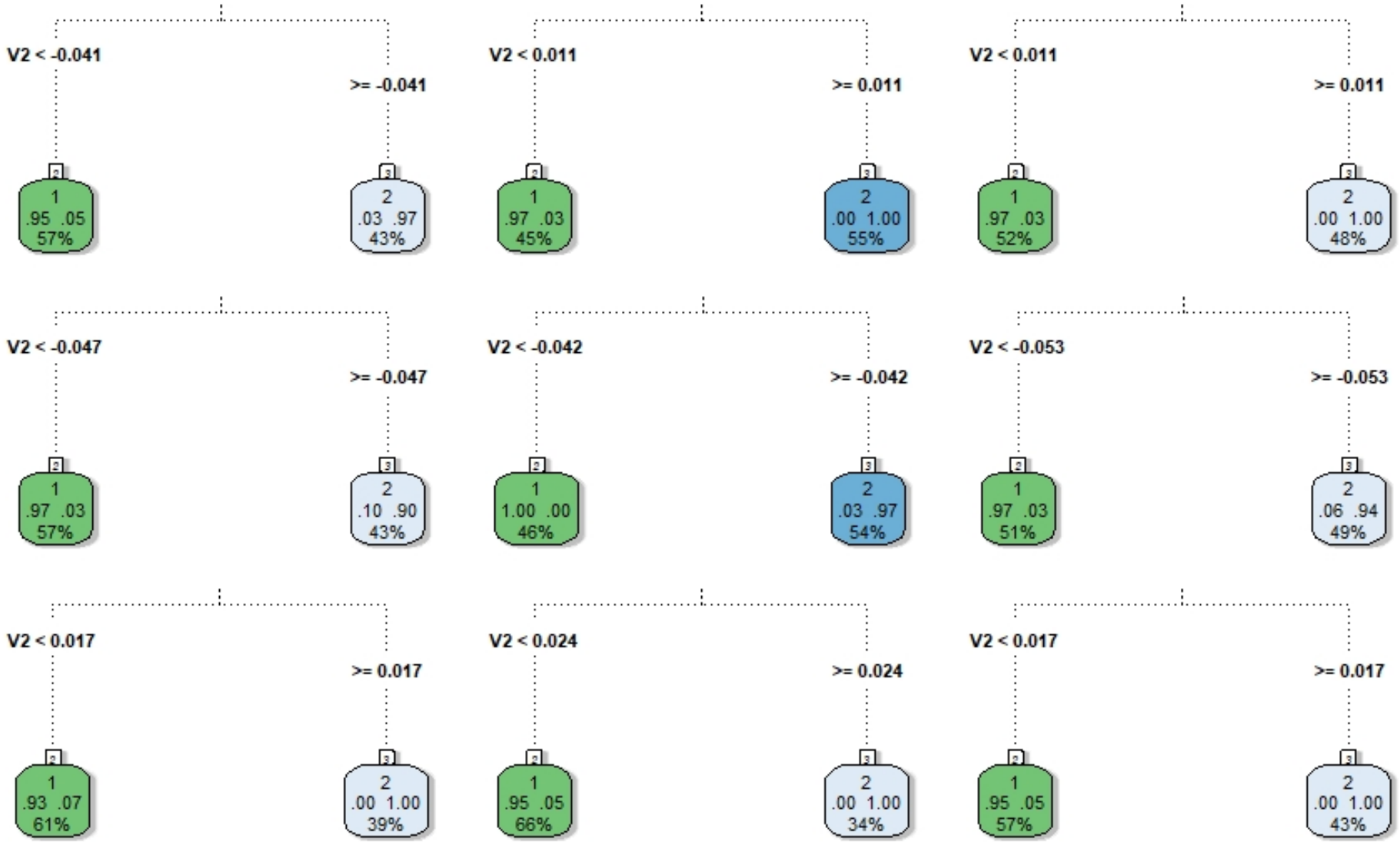
Functional means according to the split on the 2nd FPC





Functional Bagging

OOB estimate of error rate: 0.074



Functional Random Forest

At each split, three out of ten FPCs are sampled.
OOB estimate of error rate: 0.059.

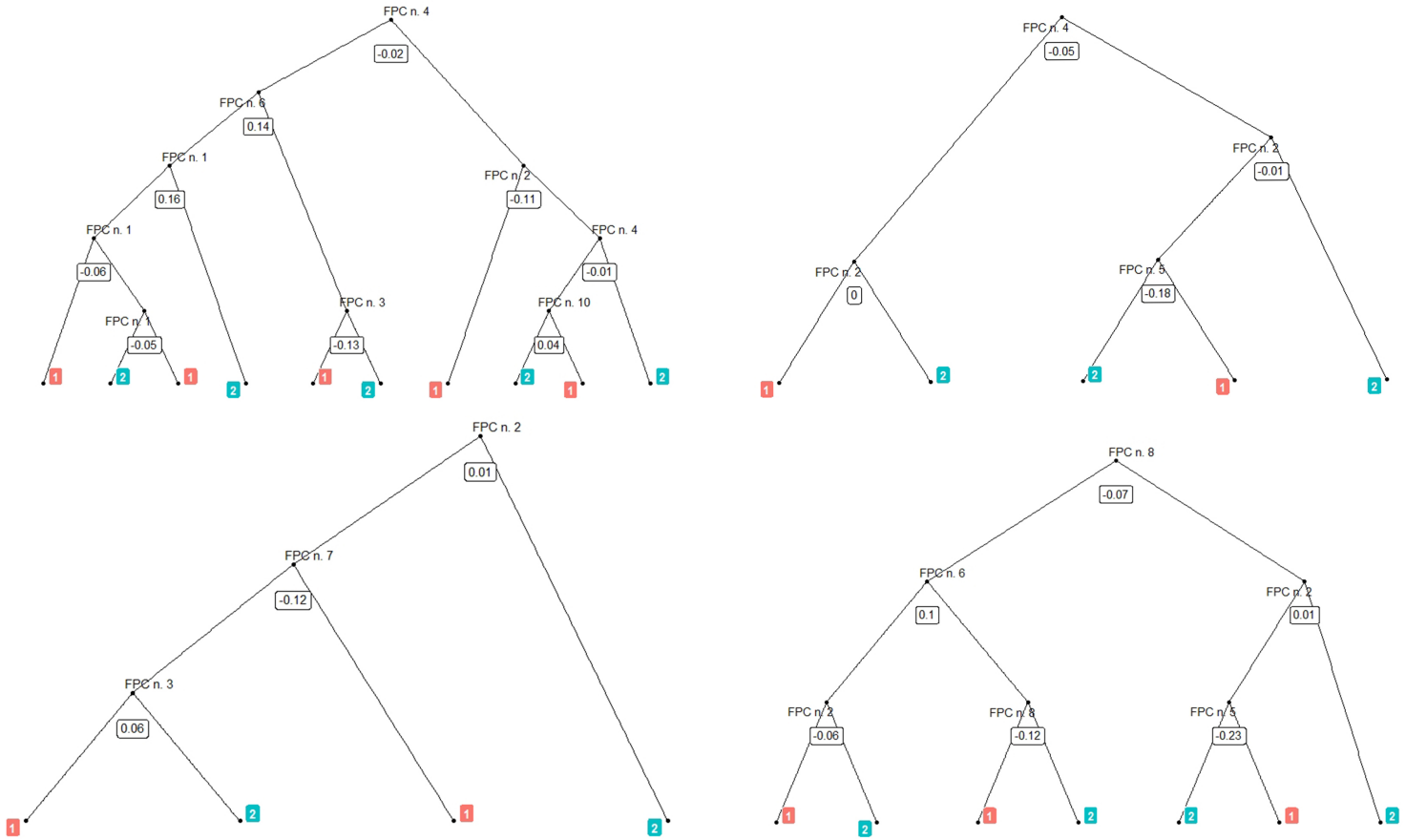


Figure: First four FDTs of the forest

Functional Random Forest Results

```
Confusion matrix:
   1  2 class.error
1 32  2  0.05882353
2  2 31  0.06060606
```

Figure: Confusion Matrix

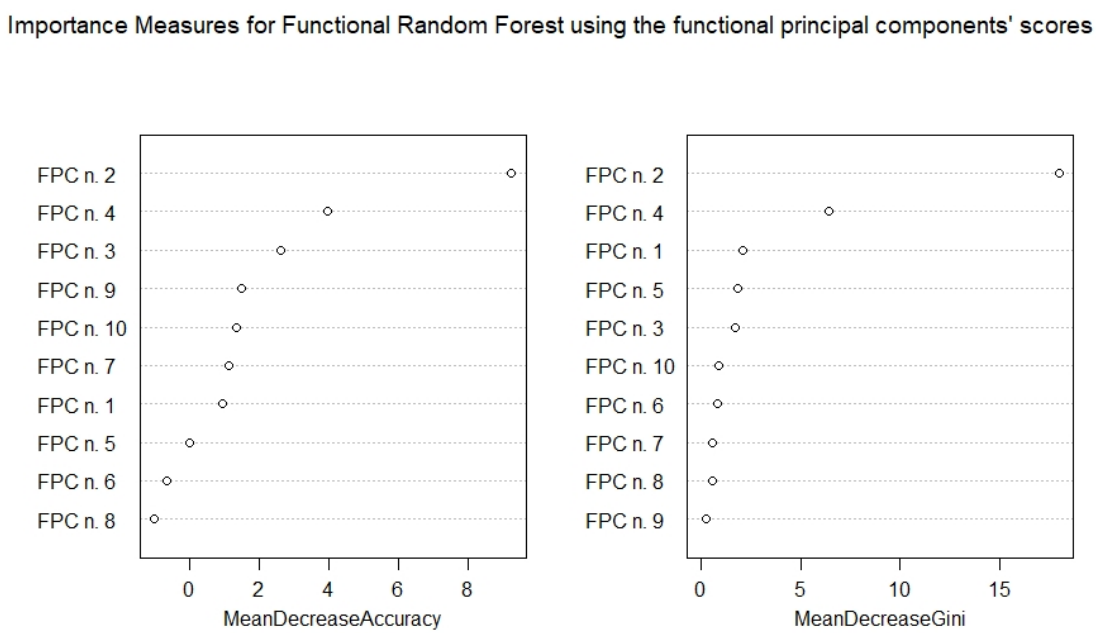


Figure: Variables Importance

Conclusions

- **FDA** provides a powerful solution to **classify high-dimensiona data**;
- FPCs are useful **tools to be considered as features** in FRF;
- This approach can be **extended to derivatives** to get more features for classification purposes;
- This approach can be **extended to b-splines and wavelets**.
- The **interpretation of the classification rule can be done according to the parts of the domain** explained by FPCs;
- This approach can be applied to environmental data, in particular to real-time streaming environmental data;
- Future developments will focus on building a **consensus FDT**.

References

Cuevas, A., 2014. A partial overview of the theory of statistics with functional data. Journal of Statistical Planning and Inference 147, 1–23. URL: https://doi.org/10.1016%2Fj.jspi.2013.04.002, doi:10.1016/j.jspi.2013.04.002.

Febrero-Bande, M., de la Fuente, M., 2012. Statistical computing in functional data analysis: The r package fda.usc. Journal of Statistical Software, Articles 51, 1–28. URL: https://www.jstatsoft.org/v051/i04, doi:10.18637/jss.v051.i04.

Ferraro, M.B., Giordani, P., 2015. A toolbox for fuzzy clustering using the r programming language. Fuzzy Sets and Systems 279, 1–16. URL: https://doi.org/10.1016%2Fj.fss.2015.05.001, doi:10.1016/j.fss.2015.05.001.

Ferraty, F., Vieu, P., 2006. Nonparametric functional data analysis. Springer, New York.

Fortuna, F., Maturo, F., Di Battista, T., 2018. Clustering functional data streams: Unsupervised classification of soccer top players based on google trends. Quality and Reliability Engineering International 34, 1448–1460. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.2333, doi:10.1002/qre.2333, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/qre.2333.

Ramsay, J., Silverman, B., 2005. Functional Data Analysis, 2nd edn. Springer, New York.

THANKS FOR YOUR ATTENTION