# Interpretable Dimension Reduction for Classifying Functional Data

TIAN SIVA TIAN * GARETH M JAMES †

**Abstract**

Classification problems involving a categorical class label $Y$ and a functional predictor $X(t)$ are becoming increasingly common. Since $X(t)$ is infinite dimensional, some form of dimension reduction is essential in these problems. Conventional dimension reduction techniques for functional data usually suffer from one or both of the following problems. First, they do not take the categorical response into consideration, and second, the resulting reduced subspace may have a complicated relationship with the original functional data. In this paper we propose a dimension reduction method, "Functional Adaptive Classification" (FAC), specifically designed for functional classification problems. FAC uses certain complexity constraints to ensure that the reduced subspace has an easily interpretable relationship to the original functional predictor. Extensive simulation studies and an fMRI (functional Magnetic Resonance Imaging) study show that FAC is extremely competitive in comparison to other potential approaches in terms of both classification accuracy and model interpretability.

**KEY WORDS:** Functional data; Classification; Dimension reduction; Stochastic search; Variable selection.

## 1 Introduction

Functional data analysis (FDA) deals with curves, or functions in general. The FDA approach [1] treats the entire function as the unit of observation, as opposed to traditional multivariate analysis which works with a vector of measurements. Dimension reduction is an important issue in the FDA literature because functional data intrinsically are infinite dimensional. There has been a great deal of research devoted to dimension reduction in high-dimensional multivariate data. Current multivariate methods generally can be categorized into three major categories: *variable selection* which selects a subset of important variables (e.g., the Lasso [2, 3], SCAD [4], nearest shrunken centroids [5], the Elastic Net [6], Dantzig selector [7], VISA [8] and FLASH [9]), *variable combination* which forms linear transformations of the variables (e.g., principal component analysis [10], partial least squares regression [11], and multidimensional scaling [12]), and some combination of *variable selection* and *variable combination* (e.g., MASS [13]).

*Department of Psychology, University of Houston, Houston, TX 77204, siva.tian@times.uh.edu

†Department of Information and Operations Management, University of Southern California, CA 90089, gareth@usc.edu

Dimension reduction in functional data has focused mainly on regression type problems [see, for example 14, 15, 16, 17, 18, 19, 20]. Relatively few papers have been written on the topic of dimension reduction for functional classification problems, even though it is as important as its regression counterpart. Dimension reduction methods for functional classification problems include generalized singular value decomposition [21], linear discriminant analysis [22], the generalized linear model [23], nonparametric methods [24, 25], a factorial method of testing various sets of curves [26], and methods based on data depth [27, 28, 29].

This paper considers dimension reduction for the functional classification problem associated with a categorical response, $Y$, and a functional predictor, $X(t)$. In practice $X(t)$ can be expressed as a $n$-by-$m$ matrix indicating $n$ observed functions measured at $m$ discrete time points. Let $X_i(t)$ denote the $i$th observed function and $Y_i$ its corresponding categorical outcome. For the sake of notational simplicity, we drop the subscript $i$ where the meaning is clear. We express a linear dimension reduction for the functional predictor as,

$$Z_j = \int X(t) f_j(t) dt, \qquad j = 1, \ldots, p, \tag{1}$$

where $Z_j$ is the $j$th dimension of the $p$-dimensional reduced subspace, $\mathbf{Z}$, and $f_j(t)$ is the $j$th *transformation function*. Equation (1) transforms the original functional predictor to a lower-dimensional multivariate subspace through the $p$ transformation functions, $f_1, \ldots, f_p$.

The two most common ways to estimate the $f_j(t)$'s are via *functional principal component analysis* (FPCA) [21, 24, 23] and *basis representation* [22] (also known as the filtering method). As with standard PCA, FPCA generates a set of orthogonal functions such that $f_1$ explains the most variance in $X(t)$, $f_2$ explains the next most variance subject to an orthogonality constraint, etc. The filtering method uses the coefficients of a basis function instead of the original measurements. The $f_j$'s then become some linear combination of the basis functions. Both approaches are fairly easy to implement. However, they have two distinct disadvantages. First, they are *unsupervised* methods, so do not utilize the response, $Y$, to solve for $f_j(t)$. As a consequence, they become less satisfactory when the goal is classification. Second, the generated $f_j$'s are usually complex functions, making it harder to interpret the relationship between the original functional space, $X(t)$, and the reduced subspace, $\mathbf{Z}$.

In this paper we propose a novel approach, "Functional Adaptive Classification" (FAC). Our method adaptively searches for optimal $f_j(t)$'s in terms of classification accuracy, while restricting the $f_j(t)$'s to take only simple forms e.g. piecewise constant or linear functions; therefore, ensuring an interpretable relationship between $X(t)$ and $\mathbf{Z}$. FAC has two nice properties. First, it implements a supervised dimension reduction by considering the group labels while selecting the $f_j(t)$'s. As a result, it tends to achieve superior classification accuracy to that of unsupervised methods. Second, FAC guarantees a simple relationship between $X(t)$ and $\mathbf{Z}$ which makes the model more interpretable without sacrificing classification accuracy. We concentrate on the two-class problem, where $Y = \{0, 1\}$, because this is the most common situation in many research areas. However, FAC could easily be extended to multi-class problems. We will discuss this problem in Section 4.

The rest of the paper is organized as follows: Section 2 introduces the general scheme of the FAC method and its implementation details. Simulation studies and a real world fMRI study are used to examine the performance of FAC in Section 3. Finally, a brief summary is given in Section 4 to conclude the paper.

# 2 Methodology

Given the infinite dimensional nature of functional data, a functional classification method will generally consist of two steps. The first step is some form of dimension reduction while the second step involves making a final classification based on the lower dimensional data. There has been considerable work performed previously on classification methods for low dimensional data. Hence, our focus here is on the first, dimension reduction, step.

We model the relationship between the categorical response variable, $Y$, and the functional predictor, $X(t)$, as independent conditional on the low dimensional representation, $\mathbf{Z}$, i.e.

$$\begin{aligned} &Y|(X(t),\, \mathbf{Z}) \sim Y|\mathbf{Z}, \\ &Z_j = \int X(t) f_j(t) dt, \quad j = 1, \ldots, p \end{aligned} \tag{2}$$

where $\mathbf{Z} = [Z_1, \ldots, Z_p]$ is a $p$-dimensional subspace. Equation (2) suggests that, provided that $F = [f_1, \ldots, f_p]$ is well chosen, there should be no loss in accuracy when utilizing $\mathbf{Z}$ instead of $X(t)$ for classification. Then the problem is to find a good $F$ so that the classification accuracy using $\mathbf{Z}$ is as high as possible subject to a simple relationship between $X(t)$ and the reduced subspace. Such a simple relationship makes it easy to evaluate the effect of $X(t)$ on $Y$ by examining the effect of $\mathbf{Z}$ on $Y$.

We formulate the problem as follows:

$$\begin{aligned} \hat{F} = \arg\min_F E_{X,Y}[e(\mathscr{M}_F(X), Y)], \\ \text{subject to } \gamma(F) \leq \lambda. \end{aligned} \tag{3}$$

where $\mathscr{M}_F$ is a pre-selected classification model applied to the lower-dimensional data and $e$ is a loss function resulting from $\mathscr{M}_F$. In this paper we examine four common choices for $\mathscr{M}_F$: logistic regression (LR), $k$-nearest neighbors (kNN), random forests (RF), and support vector machines (SVM). There are many options for $e$, such as misclassification rate (MCR), Gini index, and entropy. In this paper we use MCR as the loss function i.e. the $F$ is chosen to minimize MCR. The $\gamma(F)$ represents the complexity level of $F$ and $\lambda$ is a tuning parameter representing the degree of complexity. The constraint guarantees that $F$ has an appropriately simple structure. Intuitively, linear functions are simpler than nonlinear functions. In this paper we restrict $\gamma(F)$ to be have certain piecewise constant or linear structures. More details on the structure of the $f_j$'s are provided in Section 2.1. Hence Equation (3) corresponds to choosing a set of functions, $f_j(t)$, with appropriately simple structure, such as to minimize the test error rate on $e$ when using a given classifier, $\mathscr{M}_F$.

## 2.1 Constructing $f_j(t)$'s

For interpretation purposes we wish to generate $f_j(t)$'s with simple structures. Two of the simplest and most interpretable examples are piecewise constant and piecewise linear functions. Therefore,

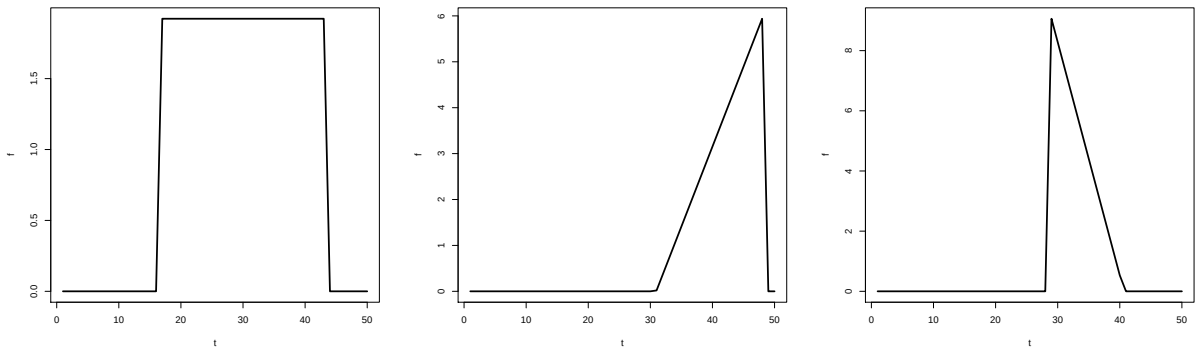
we generate each $f_j(t)$ from one of the three forms:

$$f_{ab,1}(t) = \frac{1}{b-a} I(a \leq t \leq b), \tag{4}$$

$$f_{ab,2}(t) = \frac{2(t-a)}{(b-a)^2} I(a \leq t \leq b), \tag{5}$$

$$f_{ab,3}(t) = \frac{2(b-t)}{(b-a)^2} I(a \leq t \leq b), \tag{6}$$

where $a$ and $b$ are randomly generated numbers from $\mathscr{U}(0,1)$ defining the length of the non-zero regions. Each function is standardized to have an area under the curve of 1. The function $f_{ab,1}(t)$ is constant between $a$ and $b$, $f_{ab,2}(t)$ is a linear increasing function between $a$ and $b$, and $f_{ab,3}(t)$ is a linear decreasing function between $a$ and $b$. At all other points the three functions are zero. Figure 1 shows three example of the possible forms of $f_j(t)$.




(a) Constant $f$ (b) Linearly increasing $f$ (c) Linearly decreasing $f$

Figure 1: Interpretable $f_j(t)$'s.

Even though these functions all have simple and interpretable structures, we demonstrate in Section 3 that when they are combined together they can approximate significantly more complicated relationships between $X(t)$ and $Y$. In particular when the response is modeled using a linear classifier, such as logistic regression, then $E(Y) = \mu$ can be represented as a function of a linear combination of the $Z_j$'s i.e. $g(\mu) = \mathbf{Z}^T\beta$ for some coefficient vector, $\beta$, and link function, $g$. From Equation (1) it is not hard to see that in this case $g(\mu) = \int X(t)f(t)dt$ where

$$f(t) = \sum_{j=1}^{p} f_j(t)\beta_j. \tag{7}$$

Hence, while each $f_j(t)$ provides a simple interpretation of the relationship between $X(t)$ and $Z_j$, by combining enough of these functions together we can model much more complicated relationships between $X(t)$ and $Y$. When a non-linear classifier is applied to $\mathbf{Z}$ then the relationship between $X(t)$ and $Y$ can not be summarized using a single function, $f(t)$, but the $f_j(t)$'s still provide a simple representation of each important dimension.

## 2.2 Stochastic Search Procedure

Even after specifying the simple functional forms for $f_j$ given by (4) through (6) equation (3) is still difficult to solve analytically. Stochastic search is one of the most widely used tools for solving these difficult nonlinear optimization problems. [13] proposed a stochastic search algorithm to search for an optimal transformation matrix in a multivariate setting. In this paper we generalize the stochastic search approach to the functional setting.

We first give a heuristic overview of FAC. The FAC algorithm starts with an initial $F$. At each iteration, FAC randomly generates a set of $L$ different candidate functions, $F^*$ (described in Section 2.1). The candidate functions induce the candidate subspace, $\mathbf{Z}^*$, via Equation (1). Our goal is to identify, and keep, the "good" functions and remove the others from consideration. This is achieved by using a variable selection method to choose the set of variables from $\mathbf{Z}^*$ which best predict $Y$. The $f_j(t)$'s corresponding to the selected $Z_j$ are then placed in an "historical" pool, $F_h$. In the next iteration, FAC generates new $f_j(t)$'s and joins them with $F_h$ to form a new set of candidate functions, $F^*$, and the algorithm continues as previously described. Note that, if they are useful transformations, previously selected $f_j(t)$'s should be selected multiple times. This procedure iterates until the selected subset of $f_j(t)$'s is within a given tolerance level of the previous $F$.

One would expect the most useful $f_j$'s to be selected multiple times as FAC iterates. Hence we compute an importance weight for each function included in the historical pool, $F_h$,

$$w_j = \frac{\#\text{ times } f_j(t) \text{ is selected}}{\#\text{ iterations since } f_j(t) \text{ was first generated}}. \tag{8}$$

Hence, $w_j$ measures the fraction of iterations in which our variable selection method included $f_j(t)$ in the model, since $f_j(t)$ was first generated. The $w_j$'s provide a useful measure of the most important dimensions for predicting $Y$. Using Equation (8), newer $f_j(t)$'s automatically obtain higher weights. In particular, the weight for the newest $f_j(t)$ is always 1.

If different functions are selected at each FAC iteration, $F_h$ could potentially grow large, resulting in an ultra large pool of candidate functions. This can increase the computational cost and decrease the accuracy of the variable selection method. Therefore, we restrict the size of $F_h$ to be no more than $hL$ where $0 < h < 1$. If $F_h$ grows larger than this bound, we exclude the $f_j(t)$'s with lowest importance weights, $w_j$, i.e. those functions that have been selected few times in the previous iterations. This approach works well because the solution can be unstable at the beginning of the search so FAC may select many less satisfactory $f_j(t)$'s, but as better $f_j(t)$'s are generated FAC starts to select the "good" functions in preference to the less satisfactory ones.

A more detailed overview of the FAC algorithm is presented as follows:

Step 1: Randomly generate $L$ candidate functions $F^{*(0)} = [f(t)_1^{*(0)}, f(t)_2^{*(0)}, \ldots, f(t)_L^{*(0)}]$ with the structure described in Section 2.1. Obtain the corresponding $L$-dimensional subspace $\mathbf{Z}^{*(0)}$ using Equation (1).

Step 2: In the $l$th iteration, fit a variable selection model $Y \sim \mathbf{Z}^{*(l)}$ and select the "best" $p^{(l)}$ columns from $\mathbf{Z}^{*(l)}$ based on some model fitting criterion. We denote these $p^{(l)}$ columns as $\mathbf{Z}^{(l)}$. Add the corresponding $p^{(l)}$ functions in $F^{*(l)}$ to the historical pool, $F_h$.

Step 3: Compute the importance weight, $w_j$, for each selected element of $F_h$.

Step 4: If $size(F_h) > hL$, exclude the $size(F_h) - hL$ elements corresponding to the lowest $w_j$'s.

Step 5: If convergence?

YES: Stop and output $F^{(l+1)}$

NO: Generate $L - size(F_h)$ new $f_j(t)$'s and join them together with $F_h$ to form a new candidate set, $F^{*(l+1)}$. Then repeat Steps 2 to 5 until $F^{(l+1)}$ is within a given tolerance of $F^{(l)}$.

Since Step 2 only involves a standard $L$ dimensional classification problem many standard variable selection techniques, such as the Lasso, SCAD, the Elastic Net, VISA, or FLASH, could be applied. Since the variable selection procedure involves a categorical response variable, $Y$, we implement FAC using a generalized version of the elastic net method, GLMNET [30], in a logistic regression framework to select the "best" dimensions in terms of their predictive power. GLMNET allows users to specify the mixing parameter between the $L_1$ and $L_2$ penalties used by the elastic net. We chose to set the weight on the $L_2$ penalty to zero which implements the Lasso version of GLMNET and obtains the minimum dimension size; making $\mathbf{Z}$ more interpretable. Note that there may be some situations where a slightly larger size of $F$ would produce more accurate results. In this situation one could adjust the mixing penalty parameter to place more weight on the $L_2$ penalty. In this way, the number of selected $f_j(t)$'s and the accuracy might increase, with a corresponding decrease in interpretability. We leave this possible extension for future investigation.

In Step 5, we consider the algorithm has converged if the Frobenius norm of the relative difference between the current subspace and the subspace from the previous iteration is smaller than a pre-specified threshold value. In our implementation, we declare convergence when $\|\mathbf{Z}_i - \mathbf{Z}_{i-1}\|_F / \|\mathbf{Z}_i\|_F \leq 10^{-6}$. Based on our empirical studies, usually only a few iterations are needed to reach convergence.

Once FAC selects a final set, $F$, one must apply a classification method to the resulting $p$ dimensional data. Many methods have been shown to work well on lower-dimensional data. In our simulation studies and the fMRI application we examine results from LR, kNN, RF, and SVM.

## 2.3 Tuning parameters

The FAC algorithm involves three tuning parameters; the candidate pool size, $L$, the maximum size of the historical pool, $L \times h$, and the number of dimensions in the reduced subspace, $p_l$. The computational cost increases roughly linearly with $L$ so may be high when $L$ is large. In addition, the high dimension of the candidate set may cause GLMNET to fail to correctly select the "best" set of $f_j(t)$'s. If $L$ is too small then it may not be possible to select enough $f_j(t)$'s to adequately model $f(t)$. As a consequence classification accuracy would decline.

If $h$ is too close to 1 then the candidate and historical pools will be similar. As a result few new $f_j(t)$'s will be randomly generated and the algorithm could become stuck in a sub-optimal solution. Alternatively, if $h$ is too small then potentially useful $f_j(t)$'s may be dropped from the historical pool, reducing the quality of the fit. Based on our empirical experiments on simulated and real-world applications we recommend using $L = 60\% \min\{n, m\}$ and $h = 0.8L$, where $m$ represents the number of points, $t$, that $X(t)$ is observed over. However, our sensitivity analysis in Section 3.1.4 suggests that the results are relatively insensitive to any reasonable choices for $L$ and $h$.

The number of functions selected by GLMNET at each iteration, $p^{(l)}$, varies over iterations. However, $p^{(l)}$ is automatically selected by GLMNET, based on minimizing the 10-fold cross-validated error rate. Specifically, we divide the data into 10 approximately equal parts, leave out one part each time for validation, and use the remaining parts for fitting a GLMNET model. On each of the ten fits the value for $p^{(l)}$ is chosen to minimize the error on the validation set, resulting in ten optimal values for $p^{(l)}$. We take the average of these values to obtain the final choice for $p^{(l)}$.

## 2.4 Convergence

Empirically we have found that as the search procedure iterates FAC generally converges fairly rapidly to a set of $f_j(t)$'s which give good classification accuracy. Only a few iterations are usually needed. Here we provide some theoretical motivation for the convergence of FAC.

[13] discussed conditions for convergence of the MASS approach given that an appropriate variable selection method is chosen. Although these theoretical justifications are developed in the multivariate space, they can be extend to FAC, with the added assumption that the true $f_j(t)$'s in Equation (1) can be modeled using our simple structures given by, for example, Equation (4) or (5).

Suppose a variable selection method must choose among $Z_1,\ldots,Z_L$ potential variables. Let $\mathbf{Z}_0 \in \mathbb{R}^{n\times p}$ represent the $p$-dimensional set of true variables. Note $\mathbf{Z}_0$ is not necessarily a subset of $Z_1,\ldots,Z_L$. Define $\tilde{\mathbf{Z}}_n \in \mathbb{R}^{n\times p}$ as the $p$ variables among $Z_1,\ldots,Z_L$ that minimize $\|\tilde{\mathbf{Z}}_n - \mathbf{Z}_0\|^2$ for a sample of size $n$. Then we assume that the variable selection method chosen for FAC satisfies the following assumption:

(A1) There exists some $\varepsilon > 0$ such that, provided

$$\frac{1}{n}\|\tilde{\mathbf{Z}}_n - \mathbf{Z}_0\|^2 \leq \varepsilon, \tag{9}$$

then $\tilde{\mathbf{Z}}_n$ is chosen by the variable selection method almost surely as $n \to \infty$.

Assumption (A1) is relatively mild because it essentially states that, provided our variable selection method is presented with a set of predictors which are arbitrarily close to the "true" predictors, then as $n \to \infty$, it will select the correct set of variables. Under (A1), and the assumption that the $f_j(t)$'s can be correctly modeled using our simple structures, Theorem 1 below provides some theoretical justification for the FAC method.

**Theorem 1** *Let $\tilde{\mathbf{Z}}_n^{(I)}$ represent the p variables selected by FAC after performing I iterations on a sample of size n. Then under the linear subspace model given by (1) and (2), provided a variable selection method is chosen such that (A1) holds, as n and I approach infinity,*

$$\frac{1}{n}\|\tilde{\mathbf{Z}}_n^{(I)} - \mathbf{Z}_0\|^2 \to 0 \quad a.s.$$

The proof of this result is similar to that given in [13] and hence is omitted here.

# 3 Applications

In this section, we demonstrate the advantages of FAC in extensive simulation studies and an fMRI time course study.

## 3.1 Simulation Study 1

### 3.1.1 Data generation

In each simulation scenario we first generated $n = 100$ predictor functions $X(t) = \mathbf{b}(t)^T\theta$, where $\mathbf{b}(t)$ is a $q$-dimensional natural cubic spline basis and the coefficient vectors, $\theta$, were sampled from the standard normal distribution. Each function was sampled at $m = 50$ evenly spaced points in $\boldsymbol{T} = [0,\ 1]$. At each time point we added a random measurement error. We considered four simulation settings, corresponding to two different basis dimensions, $q = 3$ and $q = 10$, and linear versus non-linear relationships between the response and the predictor. When $q = 3$, the natural cubic spline is a simple curve. In contrast, when $q = 10$, the natural cubic spline has 7 knots, and is more complex with many wiggles and jumps. The reason we chose these two values for $q$ is that we want to examine the degree to which the shape of $X(t)$ affects the estimation accuracy of the $f$'s.

In the linear setting, the response, $Y$, was generated from a standard logistic regression model,

$$\Pr(Y = 1|X(t)) = \frac{e^{\int X(t)f_0(t)dt}}{1 + e^{\int X(t)f_0(t)dt}}, \tag{10}$$

where $f_0(t)$ had the same basis as $X(t)$, i.e. $f_0(t) = \mathbf{b}(t)^T\beta$, and the coefficient vector, $\beta$, was randomly sampled for each training data set. In the nonlinear setting, $Y$ was generated from a nonlinear logistic regression model,

$$\Pr(Y = 1|X(t)) = \frac{e^{\phi(Z_1,\ldots,Z_q)}}{1 + e^{\phi(Z_1,\ldots,Z_q)}}, \tag{11}$$

where $\phi(\cdot)$ was a non-linear, non-additive function, $Z_j = \int X(t)f_{j,0}(t)dt$, $f_{j,0}(t) = \mathbf{b}^T(t)\beta_j$, and the $\beta_j$'s were randomly sampled for each training data set. Note that in both the linear and nonlinear settings the responses were generated using $f_j(t)$'s that had a different structure from that used by FAC. This corresponded to a harder real world setting where the true relationship structure is unknown.

### 3.1.2 Counterpart methods

The most commonly applied approach to functional classification problems is the filtering method, also called *feature extraction* in the engineering domain. Filtering methods work by modeling the predictor as being generated from a predefined $p$-dimensional basis i.e. $X(t) = \mathbf{b}(t)^T\theta$. One then uses the observed values of $X(t)$ to form a least squares estimate for $\theta$ and treats the estimated $\theta$ as the observed predictor. Since, in our simulation study, $X(t)$ is generated from a basis function and the response is simply a function of the basis coefficients our setup is perfectly set up for the filtering approach.

Hence, we compare the accuracy of FAC against three different implementations of the filtering method.

1. *True theta:* The ideal classification model would use the true coefficient vector, $\theta$, used to generate $X(t)$. This method represents a gold-standard that could not be achieved in practice because $\theta$ is unobserved. However it provides a useful benchmark to compare other methods against.

2. *Estimated theta from* $\mathbf{b}(t)$*:* The next best classifier would use the least squares estimate $\hat{\theta} = (B^T B)^{-1} B^T X$ where $B$ is an $m$ by $q$ matrix with each row corresponding to the true basis, $\mathbf{b}(t)$, evaluated at each of the $m$ time points that $X(t)$ is observed at. This classifier could not be used in practice because $\mathbf{b}(t)$ is unknown but again provides a benchmark to compare with.

3. *Estimated theta from a different basis:* A real world implementation of the filtering method requires the choice of a basis. In this setting one would use the least squares estimate $\tilde{\theta} = (\Phi^T \Phi)^{-1} \Phi^T X$ where $\Phi$ is an $m$ by $p$ matrix based on the chosen basis function. $\Phi$ will almost certainly be different from $B$. In this study, we used a 5-dimensional Fourier basis to form $\Phi$. This basis is often applied in practice, so $\tilde{\theta}$ can be considered as a realistic competitor to FAC.

We also compare FAC with a functional classification method based on data depth (DD) [29]. By using the depth values to represent the data, this method transforms the functional data to a 2-dimensional scatter plot, called the depth vs depth plot (DD-plot). Given the two data sets, the DD-plot is the plot of the depth values of each point from the combined set, relative to the contours of set 1 and relative to the contours of set 2. If both data sets are from the same distribution, we would expect to see a 45 degree line. Changes in the relationship between the two distributions will result in changes in the DD-plot. It has been shown that different distributional differences, such as location, scale, skewness or kurtosis differences, are associated with different graphic patterns in the DD-plot [31]. Therefore, one can find a separating curve to separate the two groups in the DD-plot. The nonparametric separating curve is obtained by minimizing the classification error in the DD-plot. As many measures of depth can be applied, we use the random Tukey depth [32] in our study.

### 3.1.3 Results

For each of our four simulation settings we implemented FAC and the filtering methods using four different classifiers: logistic regression (LR), $k$-nearest neighbors (kNN) with $k = 5$, random forests (RF) with 2000 trees, and support vector machines (SVM) with a radial basis. One would presume that, LR would perform best in the linear setting, while the other classifiers would be superior in the nonlinear cases. For each setting we generated 20 training and test data sets respectively. Each test set consisted of 1000 observations. We report the average misclassification rate (MCR) and standard errors over the 20 test sets.

In addition to the three filtering methods, $\theta$, $\hat{\theta}$, and $\tilde{\theta}$, and DD we also compared FAC to the "$X(t)$" classifier resulting from using all $m = 50$ time points of each function as the predictors. The classification results are shown in Table 1. The number in each cell is the average test error over the 20 test sets, and the number in the parenthesis is the associated standard error. We also list the Bayes rate as a reference. The Bayes rate gives a statistical lower bound on the test error achievable for a given classification problem and the associated choice of features [33]. This rate is greater than zero whenever the class distributions overlap. When all class priors and class-conditional likelihoods are completely known, one can, in theory, obtain the Bayes error directly [34]. Since this is a simulated study, we are able to compute the Bayes rate.

In the linear setting, the $X(t)$ method performed worst because it ignored the functional form of the predictor i.e. the classifier it used was for multidimensional rather than functional data. There-

Table 1: Simulation 1: Average test errors. Numbers in parentheses correspond to standard errors.

| | | | Bayes | $X(t)$ | $\theta$ | $\hat{\theta}$ | $\tilde{\theta}$ | FAC | DD |
|---|---|---|---|---|---|---|---|---|---|
| Linear | $q=3$ | LR | | 0.311(0.009) | 0.139(0.003) | 0.192(0.004) | 0.293(0.010) | 0.181(0.008) | |
| | | kNN | 0.099 | 0.269(0.008) | 0.223(0.004) | 0.245(0.007) | 0.467(0.009) | 0.259(0.010) | 0.242 |
| | | RF | (0.002) | 0.229(0.009) | 0.209(0.004) | 0.226(0.005) | 0.387(0.014) | 0.254(0.012) | (0.007) |
| | | SVM | | 0.247(0.009) | 0.202(0.003) | 0.210(0.004) | 0.328(0.015) | 0.229(0.013) | |
| | $q=10$ | LR | | 0.314(0.012) | 0.130(0.004) | 0.191(0.004) | 0.307(0.012) | 0.184(0.008) | |
| | | kNN | 0.101 | 0.324(0.013) | 0.201(0.004) | 0.294(0.004) | 0.380(0.013) | 0.232(0.009) | 0.224 |
| | | RF | (0.003) | 0.287(0.012) | 0.190(0.005) | 0.254(0.005) | 0.373(0.016) | 0.206(0.010) | (0.008) |
| | | SVM | | 0.270(0.013) | 0.185(0.005) | 0.247(0.004) | 0.342(0.015) | 0.190(0.009) | |
| Nonlinear | $q=3$ | LR | | 0.393(0.010) | 0.282(0.006) | 0.300(0.006) | 0.320(0.007) | 0.297(0.007) | |
| | | kNN | 0.133 | 0.308(0.010) | 0.220(0.005) | 0.230(0.005) | 0.236(0.006) | 0.229(0.006) | 0.208 |
| | | RF | (0.002) | 0.266(0.009) | 0.187(0.005) | 0.212(0.005) | 0.233(0.006) | 0.218(0.005) | (0.007) |
| | | SVM | | 0.224(0.007) | 0.186(0.004) | 0.208(0.004) | 0.224(0.005) | 0.188(0.005) | |
| | $q=10$ | LR | | 0.410(0.012) | 0.269(0.008) | 0.380(0.010) | 0.392(0.015) | 0.303(0.009) | |
| | | kNN | 0.111 | 0.343(0.009) | 0.237(0.005) | 0.287(0.009) | 0.394(0.017) | 0.274(0.008) | 0.309 |
| | | RF | (0.002) | 0.299(0.010) | 0.205(0.004) | 0.249(0.009) | 0.389(0.014) | 0.233(0.008) | (0.006) |
| | | SVM | | 0.270(0.008) | 0.200(0.003) | 0.246(0.005) | 0.387(0.010) | 0.211(0.006) | |

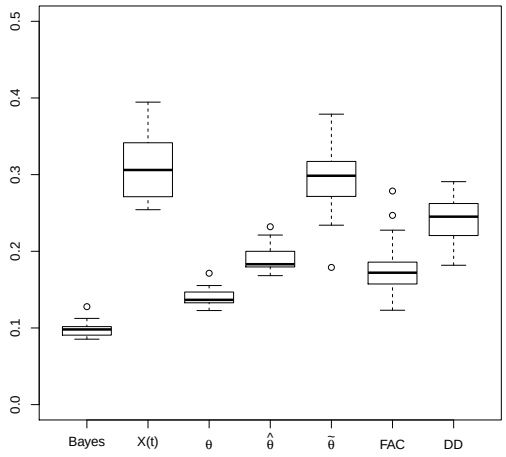
fore, ignoring the functional form of the input severely harms predictive power. As expected, $\theta$ and $\hat{\theta}$ generally produced high levels of accuracy, but these two methods can not be used in practice. FAC was quite a bit superior to $\tilde{\theta}$. Interestingly, FAC had comparable, or even slightly superior, performance relative to $\hat{\theta}$ and only performed slightly worse than $\theta$. DD generally performed better that $\tilde{\theta}$, but slightly worse than FAC and $\hat{\theta}$. In the linear setting, LR is the best classifier, and the upper two plots in Figure 2 show box plots of the test errors for the different methods using LR which was generally the best classifier in this case.

For the linear cases we illustrate the average FAC test error rate over the 20 simulation runs, for each of the four classifiers, as a function of the number of iterations, in the upper two plots in Figure 3. Note that these plots are generated purely to demonstrate the empirical convergence rate of FAC, since there is no need to perform model testing at every iteration in real-world applications. Not surprisingly, the nonlinear classifiers could not compete with LR in the linear setting. In our studies, we used 50 iterations, but the algorithm converges rapidly. After fewer than 10 iterations the test MCR started to level off, indicating a "good" reduced space had been found.

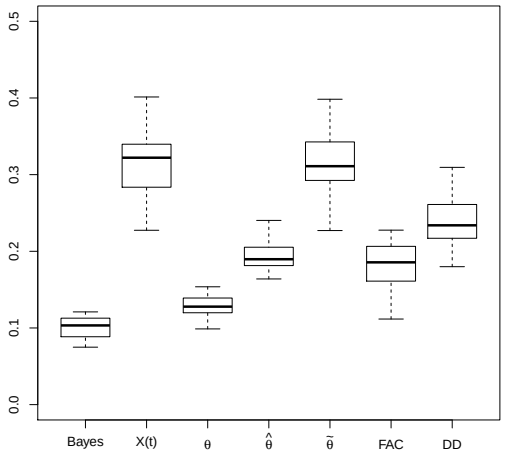
Figure 4 plots the true transformation function, $f_0(t)$ and its estimate, $\hat{f}(t)$, computed from equation (7), for iterations $4, 8, 12, 16$ and 20 of the FAC algorithm from one simulation run, respectively for $q = 3$ and $q = 10$. Even though $f_0(t)$ does not match the structure assumed by FAC, the final estimate is surprisingly accurate, even for the more complicated function where $q = 10$.

Figure 5 plots the relative "importance" weight, $w_j$, for each $\hat{f}_j(t)$, after the first 20 iterations in the $q = 3$ and $q = 10$ settings. These plots are both from one simulation run. FAC selected 5 and 23 functions for the two cases, respectively, with many of the functions chosen multiple times.

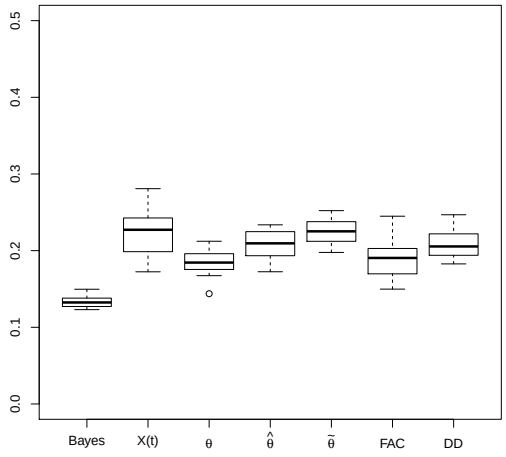
In the nonlinear setting, SVM generally gave the best predictions so we provide the MCRs using this classifier for the box plots in the bottom two plots of Figure 2. In both scenarios, FAC



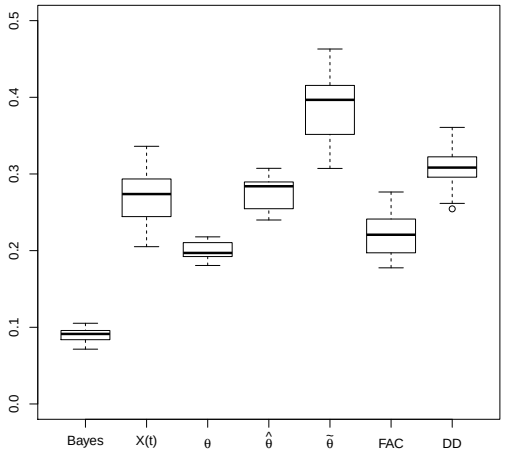

(a) Linear, $q = 3$, LR




(b) Linear, $q = 10$, LR

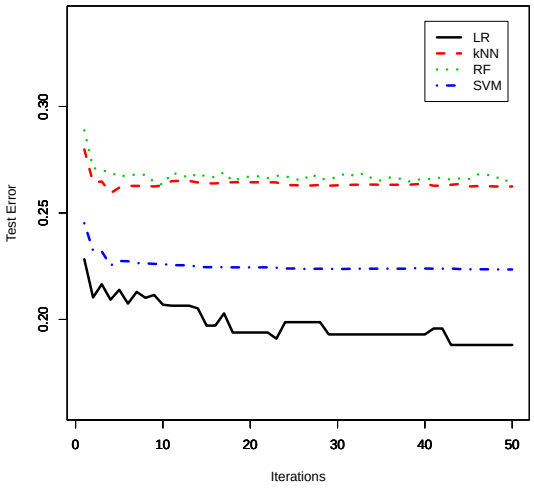



(c) Nonlinear, $q = 3$, SVM

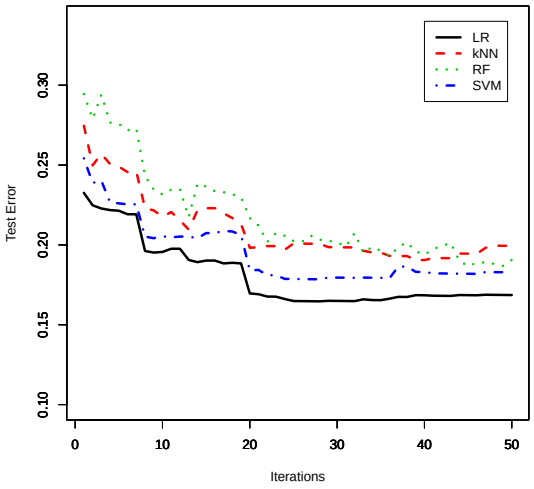



(d) Nonlinear, $q = 10$, SVM

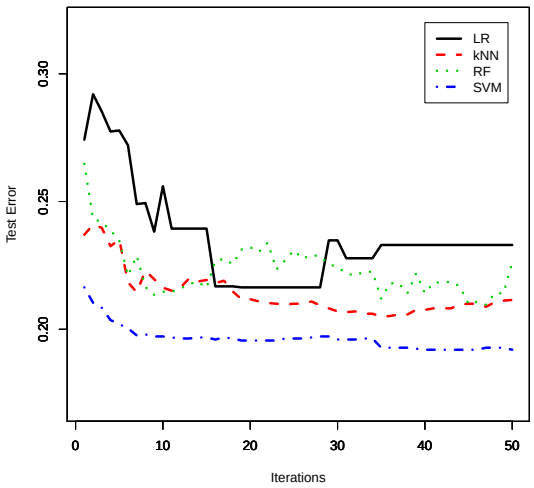
Figure 2: Box plots of the test MCR for Simulation 1. The results were produced using the LR and SVM classifiers respectively in the linear and nonlinear cases.



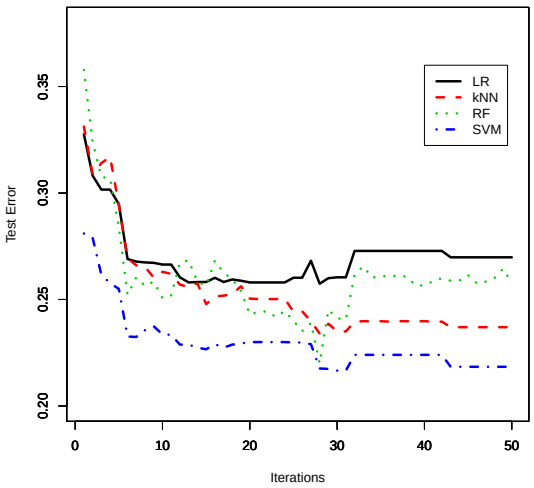

(a) Linear, $q = 3$




(b) Linear, $q = 10$

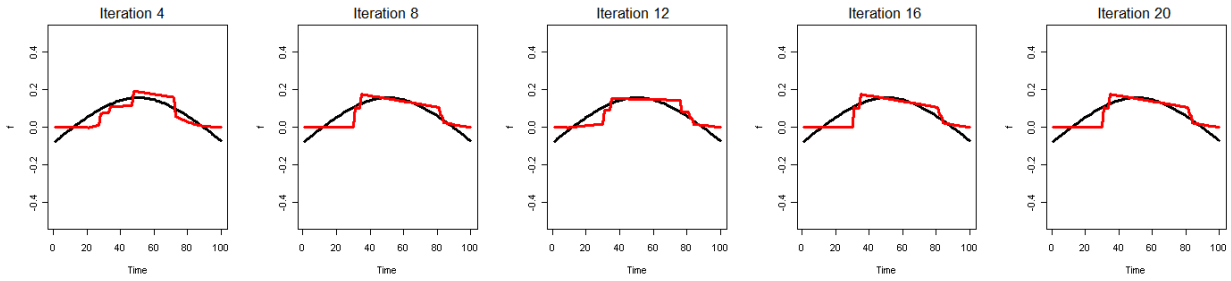



(c) Nonlinear, $q = 3$

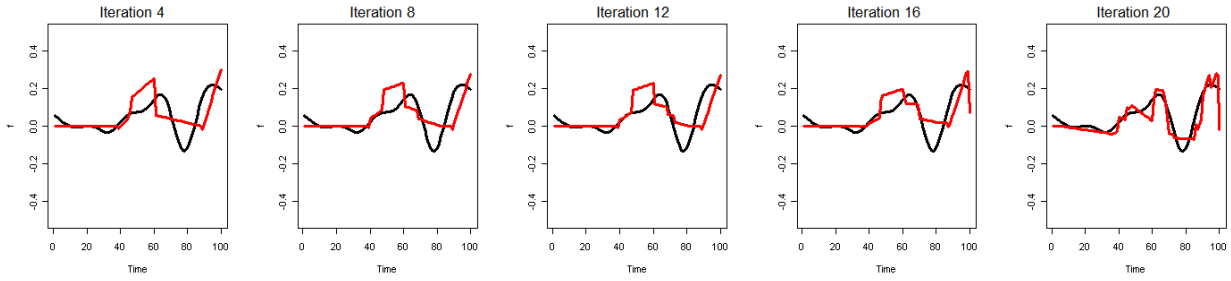



(d) Nonlinear, $q = 10$

Figure 3: Average test MCR as a function of the number of iterations for the linear and nonlinear simulations.




(a) Linear df=3




(b) Linear df=11

Figure 4: Plots of $\hat{f}(t)$ (red lines) at Iterations 4, 8, 12, 16, and 20 of FAC from a single run in the linear setting. The black lines are the true $f$.

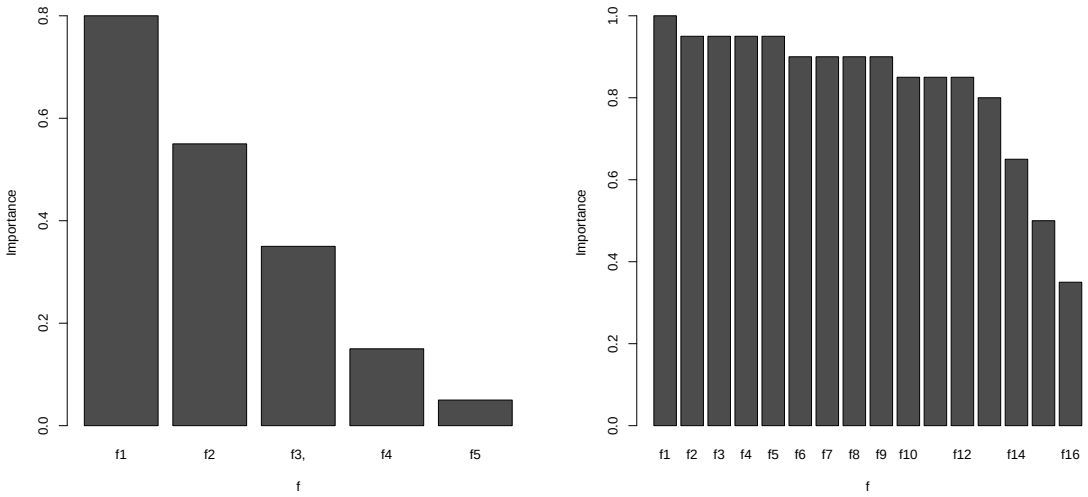
performed slightly better than $\hat{\theta}$ and significantly better than $\tilde{\theta}$. The performance of FAC was close to $\theta$. In these cases $\tilde{\theta}$ was even worse than using $X(t)$ directly, suggesting that the filtering method may be more sensitive to the choice of basis in the nonlinear setting. The relative performance of LR decreased significantly due to the nonlinear relationship between $Y$ and $\mathbf{Z}$. We also illustrate the test MCR of FAC as a function of the number of iterations in the bottom two plots of Figure 3. As with the linear situation, fewer than 10 iterations are required to achieve good classification accuracy.

### 3.1.4 Sensitivity Analysis

In this section we further explore the effects of the two major tuning parameters: the size of the candidate pool, $L$, and the size of the historical pool, $h$, using the same simulation settings as the two linear cases described in the previous section.

We chose $L$ from $(0.4m,\ 0.6m,\ m,\ 2m,\ 4m)$, and $h$ from $(0.2,\ 0.5,\ 0.8,\ 0.9)$. Since $m = 50$ this implied that $L = (20,\ 30,\ 50,\ 100,\ 200)$. We generated 20 training and test data sets and computed the average test MCR for the $5 \times 4 = 20$ possible combinations of $h$ and $L$. We plot the test MCR as a function of $L$ with different $h$ values when $q = 3$ in Figure 6 (a) and the computational cost in terms of the CPU time as a function of $L$ in 6 (b). The figures are quite similar when $q = 10$ and so are omitted here.

In general FAC performed well as long as $L$ and $h$ were not too small, with some deterioration when $h = 0.2$ or when $L = 20$. Overall, FAC was fairly stable to the choice of tuning parameters with the best results for higher values of $h$ (0.8 and 0.9) and moderate values for $L$, such as $0.6m \leq$




(a) Linear df=3 (b) Linear df=11

Figure 5: Importance weights for $\hat{f}_j(t)$'s from a single run.

$L \leq m$.

Increasing $L$ will increase the computational cost so we also investigated the CPU time for different values of $L$. Our results showed that the CPU time increased linearly with $L$. Therefore, we recommend using $L = 0.6m$ and $h = 0.8$.

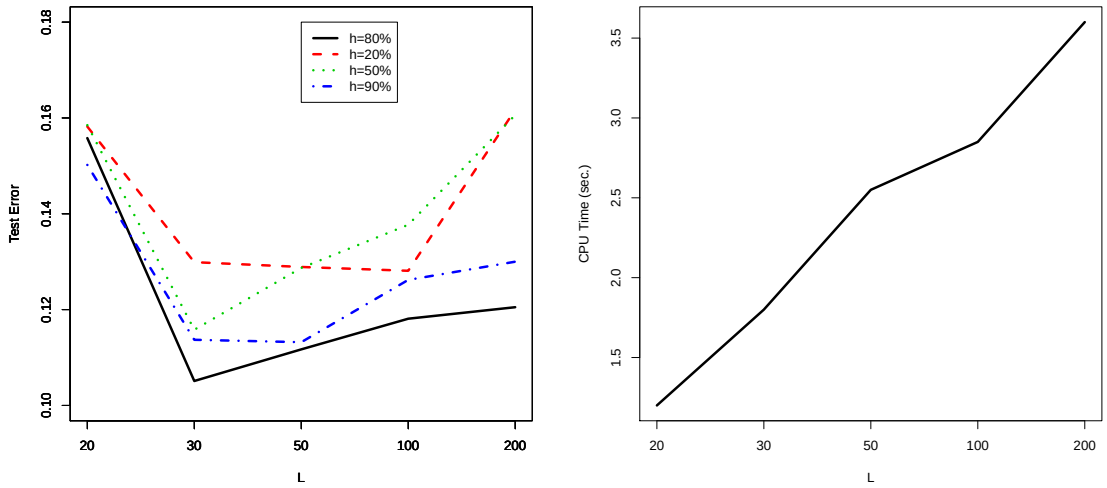
## 3.2 Simulation Study 2

In this section, we simulated an ultra-high dimensional $X(t)$ with 1000 time points. As with Simulation 1, the training and test sets contained 100 and 1000 functions, respectively. Instead of using splines as the bases to generate $X(t)$ and $f_0(t)$, we used Fourier bases with $q = 3$ and $q = 11$. We also examined the linear and nonlinear cases as with Simulation 1.

In this study, the number of time points is set to $m = 1000$. The comparison methods were $X(t)$, $\theta$, $\hat{\theta}$, $\tilde{\theta}$, and DD. The wrong basis $\hat{\theta}$ was chosen to be a B-spline with degrees of freedom 5. The average misclassification errors over 20 simulation runs in linear and nonlinear settings are listed in Table 2. As can be seen, FAC performed reasonably well. It was only slightly worse than the unrealistic $\theta$. It was as good as the unrealistic $\hat{\theta}$ in many scenarios. Figure 7 shows the estimated functions for the linear cases. FAC was still able to provide an accurate reconstruction of the shape of the true function.

## 3.3 Simulation Study 3

In this section, we simulated a scenario where $f_0(t)$ was generated jointly from natural cubic splines and Fourier bases. The true $f_0(t)$ contained three pieces. The first and the third pieces of $f_0(t)$ were generated from Fourier bases with 3 bases and the middle piece was generated from a natural cubic spline with degrees of freedom 3. The three functions did not smoothly join together. The $X(t)$ was generated from a B-spline with 5 degrees of freedom. Since the true basis of this




(a) $L$ vs. $h$ (b) Computational cost

Figure 6: Tuning parameters and computational cost in Simulation 1, linear, $q = 3$.

scenario is unknown, only three comparison methods, $X(t)$, $\tilde{\theta}$ and DD, are available. We chose a natural cubic spline with 5 degrees of freedom to generate $\tilde{\theta}$. The results are provided in Table 3. FAC dominated the other methods. Figure 8 shows the true and estimated curves from a single run of FAC and the importance weights from selected $\hat{f}_j(t)$'s from the same simulation run. FAC provides a reasonable reconstruction of the general shape of the curve but misses some information in the middle piece. Fourteen functions were selected from this simulation run.

## 3.4 fMRI time course study

In this section, we examine the performance of FAC on a functional Magnetic Resonance Imaging (fMRI) visual cortex study. fMRI is a neuroimaing technique that measures brain activity by detecting associated changes in blood flow [35]. fMRI is based on the increase in blood flow to the local vasculature that accompanies neural activity in the brain. This results in a corresponding local reduction in deoxyhemoglobin because the increase in blood flow occurs without an increase of similar magnitude in oxygen extraction. The fMRI data are a set of time series of image volumes. The brain is divided into a 3-dimensional grid of small voxels (usually about $1 \times 1 \times 1$ mm to $3 \times 3 \times 3$ mm in size). The Blood Oxygenation Level-Dependent (BOLD) response is recorded for each voxel during a certain task.

The fMRI data we used in this section are from the imaging center at the University of Southern California. They were obtained from a single male subject participating in a visual experiment. There are five subareas in the visual cortex of the brain; the primary visual cortex (striate cortex or V1) and the extrastriate visual cortical areas (V2, V3, V4, and V5). Among them, V1 is specialized for mapping the spatial information in vision, that is, processing information about static and moving objects. V2-V5 are used to distinguish fine details of the objects. Identifying these subareas of the visual cortex can help researchers understand the visual process. However, some subareas have ambiguous boundaries, so distinguishing these subareas accurately has become an important

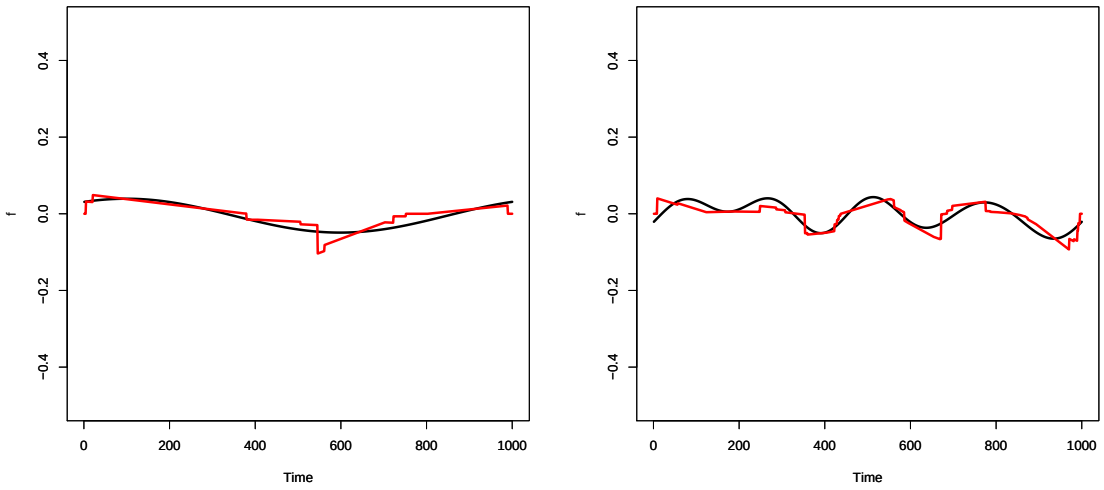
Table 2: Simulation 2: Average test errors. Numbers in parentheses correspond to standard errors.

| | | | Bayes | $X(t)$ | $\theta$ | $\hat{\theta}$ | $\tilde{\theta}$ | FAC | DD |
|---|---|---|---|---|---|---|---|---|---|
| Linear | $q=3$ | LR | | 0.451(0.003) | 0.140(0.005) | 0.147(0.005) | 0.285(0.009) | 0.152(0.008) | |
| | | kNN | 0.110 | 0.398(0.008) | 0.179(0.004) | 0.189(0.006) | 0.350(0.012) | 0.239(0.009) | 0.325 |
| | | RF | (0.003) | 0.386(0.007) | 0.138(0.003) | 0.167(0.004) | 0.359(0.012) | 0.228(0.010) | (0.009) |
| | | SVM | | 0.389(0.009) | 0.141(0.004) | 0.165(0.004) | 0.306(0.013) | 0.164(0.009) | |
| | $q=11$ | LR | | 0.465(0.003) | 0.112(0.004) | 0.124(0.005) | 0.259(0.011) | 0.118(0.005) | |
| | | kNN | 0.112 | 0.392(0.010) | 0.259(0.007) | 0.318(0.005) | 0.385(0.012) | 0.291(0.009) | 0.358 |
| | | RF | (0.003) | 0.299(0.010) | 0.200(0.005) | 0.229(0.006) | 0.319(0.013) | 0.233(0.010) | (0.010) |
| | | SVM | | 0.317(0.011) | 0.180(0.006) | 0.264(0.006) | 0.329(0.013) | 0.206(0.009) | |
| Nonlinear | $q=3$ | LR | | 0.479(0.003) | 0.312(0.007) | 0.383(0.007) | 0.412(0.004) | 0.357(0.008) | |
| | | kNN | 0.121 | 0.313(0.010) | 0.217(0.006) | 0.324(0.006) | 0.392(0.007) | 0.243(0.007) | 0.345 |
| | | RF | (0.003) | 0.256(0.008) | 0.194(0.005) | 0.288(0.004) | 0.358(0.007) | 0.214(0.008) | (0.009) |
| | | SVM | | 0.233(0.007) | 0.201(0.005) | 0.297(0.005) | 0.377(0.007) | 0.217(0.008) | |
| | $q=11$ | LR | | 0.453(0.003) | 0.298(0.008) | 0.346(0.010) | 0.397(0.015) | 0.312(0.009) | |
| | | kNN | 0.116 | 0.373(0.009) | 0.241(0.005) | 0.367(0.009) | 0.337(0.017) | 0.256(0.008) | 0.329 |
| | | RF | (0.003) | 0.365(0.010) | 0.205(0.004) | 0.287(0.009) | 0.329(0.017) | 0.236(0.012) | (0.010) |
| | | SVM | | 0.331(0.008) | 0.201(0.005) | 0.296(0.006) | 0.322(0.010) | 0.231(0.009) | |

issue. A commonly used method is to roughly identify the location and shape of an area based on clinical experience, and then draw the boundaries by hand. This method is very inaccurate and a more systematic method is needed. It is known that V1-V5 react differently to visual stimuli. That is, when a patient is conducting a visual task, the observed brain waves (functions) from voxels in V1 are different from those of V2-V5. Hence, based on the observed brain waves for a particular voxel over time it is possible to classify the voxel as V1 or not V1.

In this study, a patient was asked to conduct a visual task that included 4 trials with each lasting 101 seconds. There were resting periods before and after each trial. His BOLD response was measured every second, at each of $11,838$ voxels in the brain. In other words, there was a BOLD amplitude function over time for each voxel, $X_i(t)$, where $i$ refers to the $i$th voxel. We were interested in identifying the visual cortex, particularly which voxels corresponded to the V1 and the V3 subareas. The BOLD response function, $X_i(t)$, for a voxel in V1 should be different from a corresponding curve in V3. We take these two subareas because they are well-defined and the boundary between them is relatively clear based on empirical studies. Consequently we are almost certain of the correct group label for each function, i.e., the $Y_i$ for each $i = 1,\ldots,n$, is clear. Therefore, it is a good example to evaluate the proposed method relative to conventional methods.

Since the single trial fMRI time courses are notoriously noisy, the MCR on different trials are quite variable. Based on our experiment on this particular fMRI data, the MCR can range from 0.102 to 0.230 using FAC followed by LR. Therefore, we averaged the 4 trials to reduce noise [36] and obtained time courses with 101 measurements. Hence $X_i(t)$ was measured over 100 discrete time points while $Y_i = \{0,1\}$ indicated whether $X_i(t)$ belonged to V1 or V3. We first identified unambiguous voxels for V1 and V3 based on anatomical structure of the brain. There were 2058 such voxels, hence 2058 functions, for the V1 area. Similarly, there were 4866 voxels for the




(a) $q = 3$ (b) $q = 11$

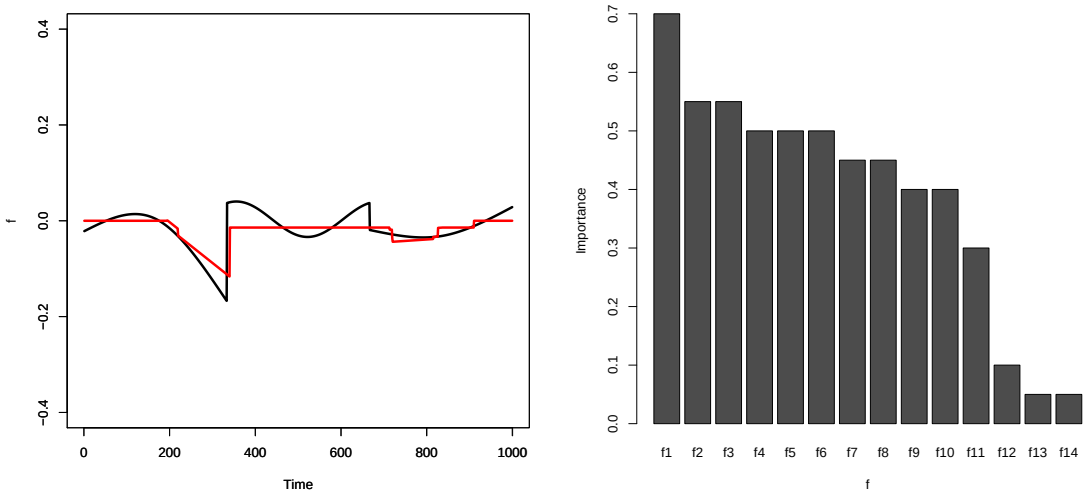
Figure 7: True and estimated functions for the linear cases in Simulation 2. The black curve is the true $f_0(t)$ and the red curve is the estimated $\hat{f}_(t)$ from one simulation run.

Table 3: Simulation 3: Average test errors. Numbers in parentheses correspond to standard errors.

| | | Bayes | $X(t)$ | $\tilde{\theta}$ | FAC | DD |
|---|---|---|---|---|---|---|
| Linear | LR | 0.100(0.003) | 0.458(0.003) | 0.410(0.004) | 0.175(0.012) | 0.357(0.013) |
| | kNN | | 0.375(0.010) | 0.368(0.009) | 0.246(0.009) | |
| | RF | | 0.343(0.010) | 0.327(0.009) | 0.213(0.008) | |
| | SVM | | 0.368(0.010) | 0.349(0.008) | 0.208(0.009) | |
| Nonlinear | LR | 0.108(0.004) | 0.437(0.003) | 0.411(0.005) | 0.295(0.009) | 0.355(0.010) |
| | kNN | | 0.392(0.009) | 0.341(0.010) | 0.343(0.011) | |
| | RF | | 0.377(0.011) | 0.352(0.009) | 0.211(0.009) | |
| | SVM | | 0.321(0.010) | 0.326(0.009) | 0.200(0.012) | |

V3 area. Since the time signals of neighboring voxels in the fMRI data are highly correlated with each other methods assuming independent white noise errors can break down. Therefore, an evenly-spaced sampling was conducted on each area to reduce the effect of correlations among neighboring voxels. Finally, we obtained a sample of 1500 voxels (functions) for each area, i.e., a total of 3000 functions. Within each area we further divided the functions into training (1000) and test (500) data. Note that we chose our training and test data from the centers of the two regions and the areas near the V1-V3 boundary, respectively. This is because the memberships are clearer in the center rather than near the boundary so using central data to train our model should produce more accurate results. Once the method is shown to be effective, it can be extended to data with ambiguous boundaries more safely.

We compared FAC to $X(t)$, DD and the filtering method previously described, using three different bases for $X(t)$: natural cubic splines ($\hat{\theta}_{\text{NCS}}$), B-Splines ($\hat{\theta}_{\text{BS}}$) and Fourier basis ($\hat{\theta}_{\text{Fourier}}$). Some preliminary examination of the counterpart methods, suggested that roughly 10-dimensional




(a) True and estimated functions

(b) Importance weights

Figure 8: Simulation 3: (a) True and estimated functions for the linear case. The black curve is the true $f_0(t)$ and the red curve is the estimated $\hat{f}_(t)$ from one simulation run. (b) The importance weights for the selected $\hat{f}_j t$'s from the same run.

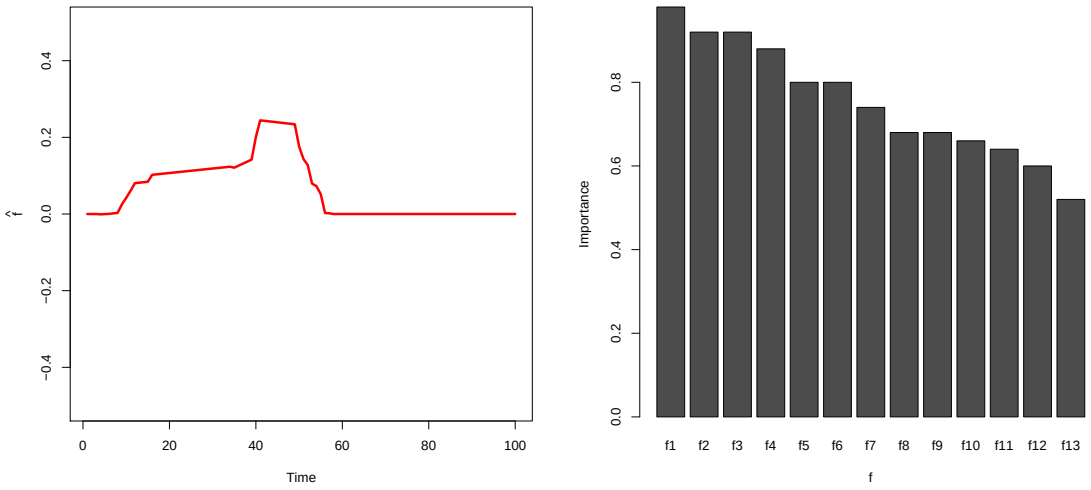
bases produced the highest accuracy levels. For both FAC and the filtering methods, we used four classifiers: LR, kNN, RF and SVM. Since FAC is based on a stochastic search we obtain slightly different results for each run. To gain a better idea of the typical average error rate we ran FAC 10 times and computed the average test MCR and the standard error. The test MCR's, displayed in Table 4, show that FAC was significantly superior to all of the filtering methods, for each of the four classifiers. It was also superior to the DD approach. The standard errors show that FAC is reasonably stable over multiple runs on this data. Figure 9 (a) shows the estimated $\hat{f}(t)$ assuming a linear relationship between the response and predictor while Figure 9 (b) plots the importance weights for the individual $\hat{f}_j(t)$'s. FAC selected 23 transformation functions and formed a step-like function. This transformation function can be interpreted as indicating that time points 20-40 of $X(t)$ have an approximately constant effect on $\mathbf{Z}$. Time points 40-50 of $X(t)$ also have a constant, but significantly larger, effect on $\mathbf{Z}$. The rest of $X(t)$ appears to have little or no effect on $\mathbf{Z}$. Figure 10 plots the predicted V1 and V3 areas from FAC using the SVM classifier. The predicted mapping matches up very closely to the mapping based on anatomical studies. Note that we do not have an absolute "truth" in this study. The "true" group labels and the "true" maps are all based on empirical and anatomical studies.

# 4 Conclusion

In this paper, we present a supervised dimension reduction method for functional data when class labels are available. Based on a stochastic search procedure, the proposed method can find a good reduced subspace for classification and improved classification accuracy. Our simulation and fMRI studies show that FAC is competitive with "ideal" methods and often superior to real world

Table 4: Test error rates and standard errors (in the parentheses) on the fMRI data for FAC and three filtering methods.

| | LR | kNN | RF | SVM |
|---|---|---|---|---|
| $X(t)$ | 0.398 | 0.371 | 0.256 | 0.221 |
| $\hat{\theta}_{\text{NCS}}$ | 0.345 | 0.258 | 0.244 | 0.237 |
| $\hat{\theta}_{\text{BS}}$ | 0.351 | 0.257 | 0.233 | 0.231 |
| $\hat{\theta}_{\text{Fourier}}$ | 0.310 | 0.218 | 0.203 | 0.199 |
| FAC | 0.119 (0.009) | 0.100 (0.007) | 0.095 (0.008) | 0.080 (0.007) |
| DD | | 0.215 | | |

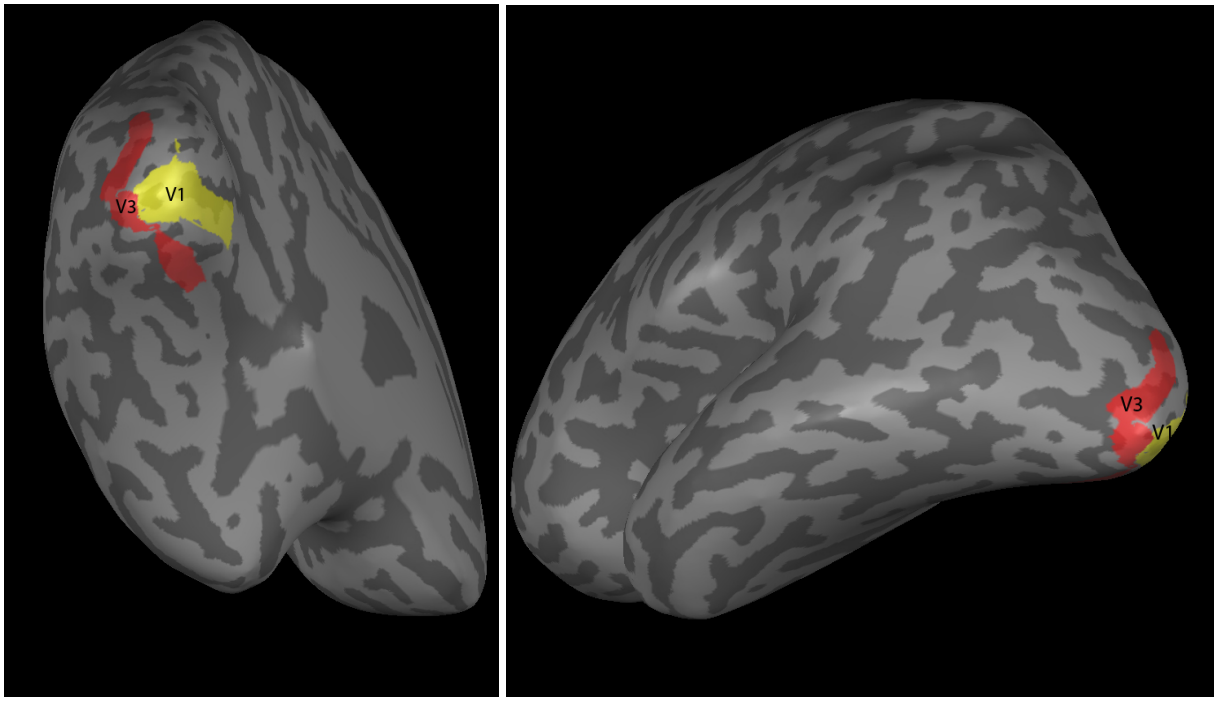



(a) $\hat{f}(t)$ (b) Importance weights

Figure 9: $\hat{f}(t)$ assuming linear relationship between the response and predictor and the importance weights for individual $\hat{f}_j(t)$.

approaches. Another advantage of FAC is that it can produce interpretable models by assigning simple structures to the transformation functions. From our simulation and real-world studies, FAC's empirical convergence rate is fast suggesting that it would be feasible to apply it to ultra-large data sets.

A number of possible extensions could be explored. First, in this study we concentrated on two class problems but FAC could easily be generalized to multi-class situations. To enable such an extension, the variable selection model in Step 2 would need to be changed to a method that considers multi-class memberships, such as the GLMNET in a framework of multinomial logistic regression. All other steps would remain the same. Second, since FAC has a stochastic fitting procedure one may obtain improved results from performing multiple fits, as we did for the fMRI data, and then selecting the best fit in terms of training error. Finally, most classification methods implicitly or explicitly assume that, conditional on group membership, the observations are independent. For example, Assumption (A1) for FAC may seem less reasonable for highly correlated data. In the case of the fMRI data the observations have a strong spatial correlation, which we remove by sub-




(a) The front view (b) The side view

Figure 10: Brain mapping of predicted V1 and V3 areas using FAC.

sampling the data. However, it would be desirable to directly incorporate this dependence structure in the methodology. A recent paper [37] uses a Markov random field approach to cluster spatially correlated functional data. This suggests a possible extension of the FAC methodology to directly model spatial correlation structures in the classification setting.

## Acknowledgement

The authors thank the editor, the associate editor, and two referees for their valuable suggestions and comments that have led to major improvement of the manuscript. This work was partially supported by NSF Grant DMS-0906784. The authors also thank the imaging center of the University of Southern California for providing the data.

## References

[1] Ramsay J, Silverman B. *Functional Data Analysis*. 2nd edn., Springer: New York, NY, 2005.

[2] Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B* 1996; **58**:267–288.

[3] Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. *Annals of Statistics* 2004; **32**:407–451.

[4] Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 2001; **96**:1348–1360.

[5] Tibshirani R, Hastie T, Narasimhan B, Chu G. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States* 2002; **99**(10):6567–6572.

[6] Zou H, Hastie T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B* 2005; **67**:301–320.

[7] Candes E, Tao T. The dantzig selector: Statistical estimation when $p$ is much larger than $n$ (with discussion). *Annals of Statistics* 2007; **35**(6):2313–2351.

[8] Radchenko P, James G. Variable inclusion and shrinkage algorithms. *Journal of the American Statistical Association* 2008; **103**:1304–1315.

[9] Radchenko P, James G. Forward-LASSO with adaptive shrinkage. *Annals of Statistics* 2010; In press.

[10] Hotelling H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 1933; **24**:417–441.

[11] Wold H. Estimation of principal components and related models by iterative least squares. *Multivariate Analysis*, Krishnaiaah P (ed.). Academic Press: New York, 1966; 391–420.

[12] Torgerson WS. *Theory and methods of scaling*. Wiley: New York, 1958.

[13] Tian TS, James GM, Wilcox RR. A multivariate stochastic search method for dimensionality reduction in classification. *Annals of Applied Statistics* 2010; **4**(1):339–364.

[14] Cuevas A, Febrero M, Fraiman R. Linear functional regression: The case of fixed design and functional response. *The Canadian Journal of Statistics* 2002; **30**:285–300.

[15] James G. Generalized linear models with functional predictors. *Journal of the Royal Statistical Society: Series B* 2002; **64**:411–432.

[16] Ferraty F, Vieu P. Nonparametric models for functional data with application in regression, time series prediction and curve discrimination. *Journal of Nonparametric Statistics* 2004; **16**(1-2):111–125.

[17] Cardot H, Sarda P. Estimation in generalized linear models for functional data via penalized likelihood. *Journal of Multivariate Analysis* 2005; **92**:24–41.

[18] Cai T, Hall P. Prediction in functional linear regression. *Annals of Statistics* 2006; **34**(5):2159–2179.

[19] Hall P, Horowitz J. Methodology and convergence rates for functional linear regression. *Annals of Statistics* 2007; **35**:70–91.

[20] James G, Wang J, Zhu J. Functional linear regression that's interpretable. *Annals of Statistics* 2009; **37**:2083–2108.

[21] Alter O, Brown P, Bostein D. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States* 2000; **97**:10 101–10 106.

[22] James G, Hastie T. Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society: Series B* 2001; **63**:533–550.

[23] Müller HG, Stadtmüller U. Generalized functional linear models. *Annals of Statistics* 2005; **33**(2):774–805.

[24] Hall P, Poskitt DS, Presnell B. A functional data-analytic approach to signal discrimination. *Technometrics* 2001; **43**(1):1–9.

[25] Ferraty F, Vieu P. Curves discrimination: a nonparametric functional approach. *Computational Statistics & Data Analysis* 2003; **44**:161–173.

[26] Ferraty F, Vieu P, Viguier-Pla S. Factor based comparison of groups of curves. *Computational Statistics & Data Analysis* 2007; **51**:4903–4910.

[27] Cuevas A, Febrero M, Fraiman R. Robust estimation and classification for functional data via projection-based depth notions. *Computational Statistics* 2007; **22**:481–496.

[28] Cuesta-Alberto A J, Nieto-Reyes A. Functional classification and the random tukey depth. practical issues. *Combining Soft Computing and Statsitical Methods in Data Analysis*, Borgelt C, González-Rodríguez G, Trutschnig W, Lubiano MA, Gil MA, Grzegorzewski P, Hryniewicz O (eds.). Springer, 2010; 123–130.

[29] Li J, Cuesta-Albertos JA, Liu RY. DD-Classifier: Nonparametirc classification procedure based on DD-plot. Available at http://personales.unican.es/cuestaj/DDPlot_Classification.pdf.

[30] Friedman J, Hastie T, Tibshirani R. Regularized paths for generalized linear models via coordinate descent. *Journal of Software* 2010; **33**(1):1–22.

[31] Liu RY, Parelius JM, Singh K. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *Annals of Statistics* 1999; **27**:783–840.

[32] Cuesta-Alberto A J, Nieto-Reyes A. The random tukey depth. *Computational Statistics & Data Analysis* 2008; **52**(11):4979–4988.

[33] Devujver PA, Kittler J. *Pattern Recognition: A Statistical Approach*. Prentice Hall: Englewood Cliffs, NJ, 1982.

[34] Fukunaga K. *Introduction to statistical pattern recognition*. Academic Press: Boston, MA, 1990.

[35] Ogawa S, Lee TM, Kay AR, Tank DW. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences of the United States* 1990; **87**:9868–9872.

[36] Thomas CG, Harshman RA, Menon RS. Noise reduction in bold-based fmri using component analysis. *NeuroImage* 2002; **17**:1521–1537.

[37] Jiang H, Serban N. Clustering random curves under spatial interdependence with application to service accessibility. *Technometrics* 2012; **54**(2):108–119.