Gene expression

Functional embedding for the classification of gene expression profiles

Ping-Shi Wu1,* and Hans-Georg Müller2,*

1Department of Mathematics, Lehigh University, Bethlehem, PA 18015 and 2Department of Statistics, University of California, Davis, One Shield Avenue Davis, CA 95616, USA

Received on June 8, 2009; revised on December 8, 2009; accepted on December 23, 2009
Advance Access publication January 6, 2010
Associate Editor: Joaquin Dopazo

ABSTRACT
Motivation: Low sample size \( n \) high-dimensional large \( p \) data with \( n \ll p \) are commonly encountered in genomics and statistical genetics. Ill-conditioning of the variance-covariance matrix for such data renders the traditional multivariate data analytical approaches unattractive. On the other side, functional data analysis (FDA) approaches are designed for infinite-dimensional data and therefore may have potential for the analysis of large \( p \) data. We herein propose a functional embedding (FEM) technique, which exploits the interface between multivariate and functional data, aiming at borrowing strength across the sample through FDA techniques in order to resolve the difficulties caused by the high dimension \( p \).

Results: Using pairwise dissimilarities among predictor variables, one obtains a univariate configuration of these covariates. This is interpreted as variable ordination that defines the domain of a suitable function space, thus leading to the FEM of the high-dimensional data. The embedding may then be followed by functional logistic regression for the classification of high-dimensional multivariate data as an example for downstream analysis. The resulting functional classification is evaluated on several published gene expression array datasets and a mass spectrometric data, and is shown to compare favorably with various methods that have been employed previously for the classification of these high-dimensional gene expression profiles.

Availability: The implementation of FEM and Classification via Functional Embedding (CFEM) as described in this article was done with the PACE package written in Matlab. The latest version of PACE is publicly accessible at http://anson.ucdavis.edu/~mueller/data/programs.html. An example MATLAB script for FEM is available at http://www.lehigh.edu/~psw205/psw205.html

Contact: psw205@lehigh.edu; mueller@wald.ucdavis.edu

1 INTRODUCTION

Functional data analysis (FDA) techniques were developed to assist scientists with modeling and analyzing curves and trajectories that can be regarded as realizations of a random function (Ramsay and Silverman, 2002, 2005). To date, FDA methodology has been used in many fields, notably the life sciences (Kirkpatrick and Heckman, 1989; Müller et al., 2008). The classification of longitudinal data with functional covariates was studied, e.g. in Müller (2005), and general issues related to the clustering and classification of functional data were discussed in Chiou and Li (2007, 2008).

The concept of functional embedding (FEM) is motivated by a need for new methodology for very high-dimensional data, and in particular the need for effective classification procedures for such data. Many efforts have been devoted to statistical methodology for multivariate data, but such methodology typically does not extend to the case of very high-dimensional data, especially if they are corrupted by noise. Owing to the problem of ill-defined inverses in high dimensions, analyzing very high-dimensional data remains a challenge. Lack of scalability to high-dimensional predictors is a feature shared by regression, correlation and discriminant analysis, all of which require some form of regularization in the high-dimensional case.

A general feature of multivariate data analysis (MDA) is that the components of the predictor vectors can be arbitrarily permuted without changing the nature of the analysis; concepts of order and neighborhood relations play no role. Pooling the effects of similarly acting predictors and averaging over their effect can be expected to remove variability and thus may lead to gains in efficiency over methods which do not take advantage of the intrinsic correlations of the components of the high-dimensional predictor vector. Taking such correlations and associated neighborhood relationships into account moves these problems closer to FDA, where topological concepts such as smoothness play an important role.

We develop here a connection between the FDA and MDA approaches for the case where the dimension of the multivariate data is large. Observing that functional data are indexed by monotone time coordinates within a compact domain, while the order of the components of high-dimensional data is arbitrary, we propose to devise an ordering of the components by means of multidimensional classical scaling, using projections into one dimension. Once the components of high-dimensional data have been ordered and embedded into a function space, FDA approaches such as functional discriminant analysis can be applied in subsequent analysis. This bridge allows to bring the whole array of FDA methodology to bear on high-dimensional data. While some theoretical support is helpful, the ultimate rationale for this approach are the successful classification results obtained for benchmark datasets of very high-dimensional gene expression measurements for moderately sized samples of patients.

A core technology for our proposal is functional discriminant analysis, and the classification of gene expression array data is our
To establish this ordering and to carry out the projection, we employ the univariate ordering, rather than the absolute values of the projections, we focus application. The FEM procedure is introduced and followed by a description of functional discrimination for classification (Section 2). Data applications are illustrated in Section 3 and concluding remarks are in Section 4.

2 METHODS

2.1 Functional embedding

Given data of dimension $p$, where $p$ is large, obtained for $n$ subjects, denote by $X = [x_1, \ldots, x_n]^T$ the input data matrix of dimension $n \times p$, where $x_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$ corresponds to the observed data vector for the $i$-th subject. Large values for the dimension $p$ of the orders 10^3 or 10^4 are, for example, encountered in microarray experiments (e.g. Amaratunga and Cabrera, 2004).

In microarray applications, $p$ corresponds to the number of genes whose expression has been recorded for each subject. The $p$-dimensional covariance vectors typically serve as predictors in a regression or classification setting.

We aim at exploiting information contained in the correlations between the predictor components and at averaging the effects of the regression coefficients that belong to highly correlated predictors, thus increasing the efficiency over an analysis which would treat every component of the predictor vector as having its own distinct regression coefficient. The tool for constructing predictor neighborhoods and exploiting smoothness is the proposed FEM. To embed the observed high-dimensional vectors into function space, we construct an ordering of the predictor components, which positions positively correlated components of the predictor vectors closer to each other, and negatively correlated components further apart on a ‘time’ axis. The ‘time’ axis is created by associating a location or ‘time’ with each of the predictor components by a projection algorithm.

To establish this ordering and to carry out the projection, we employ the classical version of multidimensional scaling (MDS) in its unidimensional form (Torgerson 1952, 1958; Gower 1966). As we utilize only the resulting univariate ordering, rather than the absolute values of the projections, we refer to this procedure as ‘ordination’.

MDS aims at finding a low-dimensional configuration of objects by retaining given pairwise distance (dissimilarity) characteristics to the largest extent possible for the projected data, minimizing one of several possible versions of ‘stress’. Special considerations apply for the unidimensional version of MDS, referred to as unidimensional classical scaling, as discussed, e.g. in section 13.8 of Borg and Groenen (2005). We use a correlation-based similarity measure.

A graphical motivation for the proposed procedures related to the gene expression application is as follows. When trying to subjectively classify the images produced in a gene expression array experiments, one often encounters difficulties. One such difficulty is the limitation of display resolution when displaying several thousands of genes expressions, as light spots will be tarnished by surrounding dark spots, obscuring underlying resolution when displaying several thousands of genes expressions, as light spots will be tarnished by surrounding dark spots, obscuring underlying patterns. Moreover, the human eye is easily misled in the assessment of light-level patterns. The proposed methods aid in overcoming these difficulties. Consider, for example, expression measurements for leukemia gene expression data. Golub et al. (1999) studied the molecular classification problem of determining the association between type of leukemia and gene expression levels. For each subject, expression levels for 7129 genes were measured. The sample contains 27 patients with acute lymphoblastic leukemia (ALL) and 11 patients with acute myeloid leukemia (AML), which are to be discriminated based on their respective gene expression patterns. There is one expression recording per gene for each subject. Selecting the 50 genes that are most significant in two-sample $t$-test comparisons, the projected data (after the ordination step), displayed in the right image of Figure 1, show differing expression patterns for the two groups, clearly more so than the original data in the left image. This provides graphical evidence of the usefulness of ordination for classification.

More formally, the FEM approach can be described as follows: given a sample of high-dimensional vectors for $n$ subjects, $x_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$,
whence, the functional version of the data becomes (FEM)

\[ X_p(t) = \sum_{k=1}^{p} \lambda_k \phi_k(t), \quad \text{if} \, \tau_k \text{for } \alpha_j, \]
\[ 1 \leq k \leq p, \quad t \geq t_{\tau_k} \text{for } \alpha_j, \quad 1 \leq \tau_k \leq p, \]
\[ t \geq \tau_k + \frac{1}{2} a_j, \quad 1 \leq \tau_k \leq p. \]

For the 'times' \( \tau \) which fall in between the constructed locations \( \tau_k \), we use linear interpolation to generate a continuous random function \( X_p \), which provides the projection of the high-dimensional data vectors into function space. The embedding algorithm may be summarized as follows.

**FEM Algorithm**

1. Calculate the sample correlation matrix \( R \) for the component variables of the high-dimensional data across the \( n \) subjects, \( \{X_{i1}, \ldots, X_{ip}\} \subseteq IR^{d \times n} \).
2. Apply unidimensional classical scaling with dissimilarity matrix \( D^2 \) and obtain the one-dimensional configuration of the variables \( \{\alpha_1, \ldots, \alpha_p\} \) for which the ranks are sorted, which are then mapped into equidistant locations \( \tau_k \) on \([-1, 1]\).
3. For each subject, obtain the embedded process \( X_p \) according to (FEM).

We assume that the embedded data \( X_p(t), \, t \in [-1, 1], \, 1 \leq \tau_k \leq p, \) obtained via (FEM), form an equidistant sampled (on the grid \( \{t_1, \ldots, t_p\} \)) realization from an underlying stochastic process \( X(t) \) with smooth trajectories. The smoothness assumption facilitates the application of current estimation techniques for Functional Principal Component Analysis (FPCA). One may also apply suitable smoothing methods to the data, which are assumed to be discretely sampled from an underlying trajectory, to create smooth continuous trajectories; this is a standard preprocessing method in FDA (Ramsay and Silverman, 2005). Among the various available smoothing techniques, we choose local polynomial smoothing (Fan and Gijbels, 1996) throughout.

Assuming that processes \( X_p \) are i.i.d. distributed as \( X_p \), with \( E(X_p) = \mu \) and \( \text{cov}(X_p(t), X_p(t)) = \text{G}(t, t) \), where we suppress indices \( p \) in \( G \) and other quantities in the remainder of this section, a representation for these processes is obtained via the linear auto-covariance operator

\[ A_p f(t) = \int f(t) G(t, s) ds, \]

defined for \( f \in L^2 \). The orthonormal eigenfunctions \( \phi_k \) of these operators satisfy \( \int \phi_j(t) \phi_k(t) dt = 1 \) for \( j = k \) and \( = 0 \) for \( j \neq k \), with ordered eigenvalues \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \cdots \). One obtains well-known representations for the covariance surface \( \text{G}(t, s) = \sum_{k=1}^{\infty} \lambda_k \phi_k(t) \phi_k(s) \) and for the random trajectories \( X_p(t) \).

\[ X_p(t) = \mu(t) + \sum_{k=1}^{\infty} \lambda_k \phi_k(t). \]

The latter is the Karhunen–Loève representation (Ash and Gardner, 1975) with functional principal component (FPC) scores \( \xi_k \). These scores are uncorrelated random variables, satisfying \( E(\xi_k^2) = 0 \), \( \text{var}(\xi_k) = \lambda_k \) and \( \sum \lambda_k = \infty \), where

\[ \xi_k = \int (X_p(t) - \mu(t)) \phi_k(t) dt. \]

In the FEM setting, the measurements for random trajectories \( X_p \) are obtained on an equidistant grid \( t_{\tau_k} = \cdots = t_{\tau_k} \) on \([-1, 1]\). The measurements may be viewed as contaminated with additional measurement errors (Shi et al., 1996; Rice and Wu, 2000; Yao et al., 2003). Observations \( Y_j \) of the random function \( X_p \) at time points \( t_j \) with additional measurement errors \( \epsilon_j \) that are assumed to be i.i.d. with finite variance \( \sigma^2 \) and independent of the random coefficients \( \xi_k, \, i = 1, \ldots, N, \, j = 1, \ldots, m_c \), can then be represented as

\[ Y_j = X_p(t_j) + \epsilon_j = \mu(t_j) + \sum_{k=1}^{\infty} \xi_k \phi_k(t_j) + \epsilon_j. \]

\[ \text{where } -1 \leq t_j \leq 1, \, k(\epsilon_j) = 0 \text{, var}(\epsilon_j) = \sigma^2. \]

The FPC scores \( \xi_1, \xi_2, \ldots, \xi_{d_k} \), suitably truncated at a finite number of included components \( d_k \), furnish a low-dimensional configuration of the high-dimensional data, while reflecting the prominent modes of variation of the functionally embedded data. To obtain the FPC scores from functional data sampled on a discrete grid \( (3) \), we first obtain smooth estimates of the overall mean function \( \mu \) and of the overall covariance function \( G \) by pooling the data of all trajectories in the training set and applying local linear smoothing techniques. Then the FPC scores are estimated by a shrinkage method. These estimates are described in detail in Yao et al. (2003).

### 2.2 Functional discriminant analysis

Classification of functional data may be based on the FPC scores which represent the observed stochastic trajectories uniquely. They may serve as treatments for any discriminant function, including classical linear, quadratic or logistic discrimination methods. Logistic discrimination, employing the FPC scores as predictors and the group labels as binary responses, was for example, used for the classification of time course gene expression data (which from the outset are in functional form) in Leng and Müller (2006).

Classification procedures typically aim at minimizing misclassification rates. A second distinct goal, especially for the classification of high-dimensional data, is variable selection, aiming at parsimonious representations of the classifier, as well as accuracy. For example, in Figure 1 only the top-ranked 50 genes, preselected by pairwise t-test comparisons, as proposed previously in Nuygen and Roche (2002) and Zou and Hastie (2005), were considered. By excluding non-discriminating predictors, the calculations involved in the classification scheme can be simplified and also possibly fewer data need to be collected. This motivates classification schemes which include variable trimming features. We consider here data which contain both training and test sets, as in the classical classification problem.

A functional classification technique that we employ is binary functional regression, a special case of the generalized functional linear model (Müller and Stadtmüller, 2005, compare also James, 2002 and Escabias et al., 2004).

In the generalized functional linear model, the conditional mean response and the functional predictors are related by

\[ E(Y|X) = \int f(t) \mu(t) dt, \]

where \( f(t) \) is a smooth monotone link function and \( \mu(t) \) is the coefficient function which determines the linear predictor. In the binary case, the responses \( Y \) are coded as 0–1 variables, indicating group membership, and the link function can be chosen as any smooth distribution function. With the orthogonal eigenfunctions \( \phi_k(t) \) of the auto-covariance operator of \( X \), one has the representations

\[ X(t) = \mu(t) + \sum_{k=1}^{\infty} \lambda_k \phi_k(t), \quad \beta = \sum_{k=1}^{\infty} \lambda_k \phi_k(t). \]

Hence, the linear predictor can be re-expressed as

\[ \eta = \int \left\{ X(t) - \mu(t) \right\} \beta(t) dt \]

\[ = \int \left\{ \sum_{k=1}^{\infty} \lambda_k \phi_k(t) \right\} \left\{ \sum_{k=1}^{\infty} \lambda_k \phi_k(t) \right\} dt = \sum_{k=1}^{\infty} \lambda_k \xi_k. \]

Specifically, we use the functional analog of logistic discriminant analysis by choosing \( g \) as the logit link for the dichotomous classification problem that we consider here.
An alternative that we implemented in our algorithm is to eliminate a fixed number of predictors \( M \) based on the outcome of CFEM obtained for different truncation parameters \( M \). This motivates variable trimming by eliminating those support points \( \tilde{t}_j, 1 \leq j \leq p \), for which \( |\hat{\beta}(\tilde{t}_j)| < \alpha \), for a prespecified threshold \( \alpha > 0 \). An alternative that we implemented in our algorithm is to eliminate a fixed fraction \( y \) of predictor variables which are associated with the smallest values of \( |\hat{\beta}(\tilde{t}_j)| \).

Incorporating either one of these trimming schemes suggests an iterative procedure, where in each iteration step FEM is followed by a trimming step, after which a new ordination is implemented for the reduced number of variables, followed by the next FEM step. This process is iterated as long as estimated misclassification error is reduced and the number of included predictors does not fall below a prespecified minimum. To summarize, the algorithm is as follows:

**Classification via Functional Embedding (CFEM)**

1. **Initialization**: preselect predictor variables, reducing the dimension of the data vectors in the training set. A common approach that we adopt in our implementation is preselection based on the significance of two-sample \( t \)-tests, comparing the mean values for each predictor variable between the two groups and retaining only the most significant predictors (for more details, see (7) below).

2. **FEM**: obtain the unidimensional configuration and ordination of the data vectors from the training set and construct the process \( X_{ip} \) for each subject in both training and test set, for currently included predictor variables.

3. **Functional Classification**: apply a functional classification method. In our implementation, this is based on the M FPC scores obtained by FPCA from the embedded random trajectories, which are predictors in a logistic regression. The apparent misclassification error rate for the training set is obtained.

4. **Thresholding**: remove a fraction \( y \), \( 0 \leq y < 1 \), of the predictor variables by removing predictors with support points \( t_j \), if \( |\hat{\beta}(t_j)| \) is below the empirical \( y \)-quantile of all values \( |\hat{\beta}(t_j)|, j = 1, \ldots, p \).

5. **Iteration**: repeat Steps 2-4 until the number of remaining variables is falling below the prespecified minimum or the apparent misclassification rate is increasing.

At the termination of the iteration, apparent misclassification error on the training set and the more reliable misclassification error on the test set are the relevant outcomes of interest, in addition to the number of predictor variables that are selected to carry out the final classification step. The auxiliary parameters in this algorithm include two smoothing bandwidths, the truncation parameter \( M \) determining the number of included FPC scores in (5), the threshold \( y \) and the prespecified minimum number of predictors (used as a stopping criterion). One of the two bandwidths is used to pre-smooth each random trajectory (so that no further smoothing is needed for the overall mean function), and the other to estimate the covariance function of the underlying stochastic process. These bandwidths can be chosen by cross-validation, minimizing the squared prediction error for the test set, by a variant such as generalized cross-validation, or simply by the rule of thumb (Silverman, 1986). The choice of \( M \) can be based on the outcome of CFEM obtained for different \( M \), in terms of misclassification error for the test set, taking also into consideration the total number of selected predictors, both of which one wants to monitor.

For those situations where one has only one set of data, and no test set is available, one can randomly bisect the data into two subsets (e.g. see Dudoit et al., 2002), applying the proposed CFEM algorithm to a number of artificial training and test sets and then targeting the average observed misclassification rate over the test sets. The choice of the threshold parameter \( y \) can be coupled with the choice of \( M \), aiming at minimizing the misclassification rate; alternatively a fast implementation is to fix it at a moderate proportion level (e.g. \( y = 0.1, 0.2 \)). For the applications in Section 3, \( y = 0.2 \) was used throughout. Moreover, the minimum number of predictors at which CFEM stops can also be set at a suitable small value, we fixed this value at 10 in our implementation.

### 3 CLASSIFICATION OF GENE EXPRESSION PROFILES AND SPECTROMETRIC DATA

Here we propose to apply CFEM, as described in Section 2. The starting point are high-dimensional gene expression vectors or profiles which are mapped into functions and eventually condensed to a few FPC scores. The task starts with preselection of a subset of gene expressions by two sample \( t \)-tests, retaining only the genes which show highly significant differences in expression between the groups to be compared. If the sample mean of the expressions of the \( j \)-th gene across the \( n_1 \) subjects in group 1 is denoted by \( \bar{x}_{1j} \), and the sample mean for the \( n_2 \) subjects in group 2 by \( \bar{x}_{2j} \), with standard deviations (SDs) \( s_{1j} \) and \( s_{2j} \), the absolute value of the \( t \)-statistic for comparing the means between the two groups for the \( j \)-th gene expression is

\[
T_j = \frac{\bar{x}_{1j} - \bar{x}_{2j}}{\sqrt{\frac{s_{1j}^2}{n_1} + \frac{s_{2j}^2}{n_2}}}, \quad 1 \leq j \leq p. \tag{7}
\]

Performing these pairwise comparisons for all \( p \) genes, and subsequently ordering the \( T_j \) in descending order, one may preselect the top \( p' \) genes, \( p' < p \), as those with the most significant values of the \( t \)-statistic. This preselection via two sample \( t \)-test is a fast and straightforward tool for initial dimension reduction and has been widely used (see e.g. Nguyen and Rocke, 2002; Zou and Hastie, 2005). Note that preselection by two sample \( t \)-tests could also be replaced by other parametric/non-parametric alternatives. This replacement does not significantly affect the outcome of FEM and CFEM, however \( t \)-tests have the advantage to be computationally fast. After the initial preselection step, the data become \( (x_{ij}, y_i) \leq i \leq n, \) where \( y_i = 0 \) for group 1 and \( y_i = 1 \) for group 2, and each \( x_{ij} = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p \) is extracted as a sub-vector from the original \( p \)-variate vector. This preselection was illustrated for the leukemia example, where we chose \( p' = 50 \) in Figure 1. Normalization is an important part of preprocessing gene expression data and is typically done by standardizing data with respect to either subject or gene. For the FEM algorithm, we found it useful to standardize the data within each gene to attain zero mean and unit SD, and this normalization scheme was implemented for all data except for the leukemia data where it was preferable to standardize the data per subject and across genes. In the following applications, we also provide the results of principal component regression (PCR; svd.glmfit in MATLAB) and partial least square (PLS; plregress in MATLAB) for comparisons with CFEM on genomic data; autoregressive modeling with linear discriminant analysis (AR; ar in R) on spectrometric data. For the choice of the number of components in PCR and PLS and of the order in AR, the empirically optimal choice (ranging from 1 to 10), yielding the lowest misclassification error was adopted.
Their data contain two subsets collected from either bone marrow or peripheral blood samples. One set has 27 ALL cases and 11 AML cases; this is the training set. The test set consists of 20 ALL cases and 14 AML cases. The gene expression data were obtained from Affymetrix chips, containing 7129 genes (see http://www.broadinstitute.org/MPR for details).

Functional embedding

3.1 Diagnosis of leukemia

Golub et al. (1999) studied the molecular classification problem to diagnose type of leukemia based on gene expression array data. Their data contain two subsets collected from either bone marrow samples or peripheral blood samples. One set has 27 ALL cases and 11 AML cases; this is the training set. The test set consists of 20 ALL cases and 14 AML cases. The gene expression data were obtained from Affymetrix chips, containing 7129 genes (see http://www.broadinstitute.org/MPR for details).

Golub et al. (1999) applied both a weighted voting scheme and a 2-cluster self-organizing map (SOM) to classify the data in the training set. The weighted voting scheme correctly classified 36 out of 38 samples, while the 2-cluster SOM produced one cluster containing 24 AML and one ALL sample and another cluster containing 10 AML and 3 ALL samples. For the test set, the weighted voting machine correctly classified 29 out of 34 samples, with 5 left undetermined. For the same data, Furey et al. (2000) used the support vector machine to correctly classify 30–32 out of the 34 samples in the test set; Nguyen and Rocke (2002) applied PLS with support vector machine to correctly classify 30–32 out of the 34 samples in the test set; Nugyen and Rocke (2002) applied PLS with support vector machine to correctly classify 30–32 out of the 34 samples in the test set; Nugyen and Rocke (2002) applied PLS with support vector machine to correctly classify 30–32 out of the 34 samples in the test set.

First, we show the effect of the ordination from FEM for these data. This effect is illustrated in Figure 2 for the initial embedding step, where \( p' = 1000 \) genes were preselected and the cross-sectional mean expression profiles were plotted before and after the ordination step of the FEM algorithm. For better visualization, the original data were normalized with respect to each gene for this graph.

One can see that the effect of the ordination is quite dramatic in bringing out structures that remain hidden when plotting the original data. Applying the proposed CFEM, we compared the results when initializing the algorithm at different numbers of genes \( p' \), preselected by the \( t \)-statistic, e.g. starting with the top-ranked 50, 100, 500 or 1000 genes. For different starting numbers \( p' \), CFEM was applied for various choices for the number of included functional components \( M \). selected from \( M = 1, \ldots, 10 \). Table 2 lists the resulting optimal misclassification error rates obtained for the test set, along with the corresponding choices of \( M \) and of the final number of selected predictor genes on which the classification results were based.

The derivation of estimated eigenfunctions and of the FPC scores is based on mean function and covariance function estimates, obtained by pooling all data from the training set after FEM. For the case where the algorithm is initialized at \( p' = 100 \) preselected predictor genes, estimated mean functions for the various groups are shown for the final iteration of the FEM algorithm in Figure 3 and are seen to differ substantially between the groups. The first four eigenfunction estimates as derived from the fitted covariance surface, obtained from the representation of \( G \) given prior to Equation (1), depicted in Figure 4, are shown in Figure 5. These functions are instrumental in constructing the FPC scores which serve as predictors for logistic classification. The FPC scores are obtained by applying Equation (2) and numerical integration. The final parameter function estimate \( \hat{\beta}(\cdot) \) is shown in Figure 6. Regions where the values of \( |\hat{\beta}(\cdot)| \) are large (especially in the right half) are those where the expressions of the reordered genes tend to differ most between the two groups.

As the results in Table 2 demonstrate, the classification results obtained for FEM are excellent for these data, with a zero misclassification rate for the test set. Moreover, the size of the gene set selected in the final step is smaller than for any of the competing approaches (Nugyen and Rocke, 2002; Zou and Hastie, 2005), the only exception being the case where one starts with 1000 preselected genes. These results are relatively robust with regard to the precise number of preselected genes as well as the number of selected components \( M \). As for the comparison with PCR and PLS,
CFEM performs best in terms of both misclassification rate and the number of included components. For reference, we also provide the cross-validation results for the training set, as obtained from 100 runs. In each run, we randomly split the training set into a set of size 30 for training purpose and another set of size 8 for testing purposes, keeping approximately the same ratio of ALL to AML as observed for the entire training set. This baseline result is provided in Table 1; we note that the real performance comparison of the methods, however, needs to be based on the test sets, as shown in Tables 2 and 4.

<table>
<thead>
<tr>
<th>Error (SEM)</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCR</td>
<td>17.13% (1.27%, 9)</td>
<td>14.12% (0.91%, 8)</td>
</tr>
<tr>
<td>PLS</td>
<td>28.13% (1.72%, 2)</td>
<td>21.5% (1.61%, 2)</td>
</tr>
<tr>
<td>CFEM</td>
<td>13.63% (1.22%, 2)</td>
<td>15.88% (1.2%, 4)</td>
</tr>
<tr>
<td>No. of genes selected</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Error (SEM)</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>PCR</td>
<td>6.5% (0.88%, 5)</td>
<td>4.75% (0.9%, 10)</td>
</tr>
<tr>
<td>PLS</td>
<td>9.63% (1.17%, 5)</td>
<td>7.02% (1.09%, 5)</td>
</tr>
<tr>
<td>CFEM</td>
<td>12.38% (1.25%, 9)</td>
<td>7.75% (0.94%, 4)</td>
</tr>
<tr>
<td>No. of genes selected</td>
<td>41</td>
<td>208</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Starting no. of genes p′</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
</tr>
<tr>
<td>PCR</td>
</tr>
<tr>
<td>PLS</td>
</tr>
<tr>
<td>CFEM</td>
</tr>
<tr>
<td>No. of genes selected</td>
</tr>
</tbody>
</table>

Table 1. Misclassification error rates for the training set, comparing PCR, PLS and CFEM, with number of included components M in parentheses, and the resulting number of selected predictor genes in CFEM, for the leukemia data of Golub et al. (1999)
3.2 Prognosis of breast cancer

van’t Veer et al. (2002) studied the prediction of clinical outcomes for breast cancer patients, based on their gene expression profiles. Gene expression was measured for primary breast tumor tissue of 117 patients without tumor cells in local lymph nodes at diagnosis. For each of the patients, an expression profile containing the expression of 24,919 genes was recorded. We adopt the same procedures for preprocessing the data. Seventy-six lymph-node-negative patients were selected to be included in the training set. Preprocessing led to expression profiles containing expression levels for 4,919 genes. Out of the selected 76 patients, 32 patients developed distant metastases within 5 years, and thus had a bad prognosis, while the other 44 patients did not relapse for at least 5 years from their initial diagnosis. Another set of data containing 19 lymph-node-negative patients (12 with poor prognosis and 7 with good prognosis) serves as test set.

We applied PCR, PLS and CFEM after preselecting genes with the $t$-statistic (7) with the goal to determine the prognosis of a patient from the gene expression profile. The classification results comparing PCR, PLS and CFEM are shown in Table 3.

For the same data, van’t Veer et al. (2002) applied supervised classification on expression profiles consisting of 70 ‘optimal’ genes and reported 2 out of 19 misclassified cases in the test set. Yeung et al. (2005) used Bayesian model averaging for the same data and selected 6 genes which resulted in 3 out of 19 misclassified cases. In comparison, CFEM selects 16 genes for which it achieves a 2/19 error rate, which is a competitive result. From the results in Table 3, one finds that CFEM also outperforms PCR and PLS. Analogously to Table 1 for the leukemia data, to establish a baseline, we provide misclassification error rates for the training set from 100 cross-validation runs in Table 4.

3.3 Lymph node status for breast tumors

West et al. (2001) studied the relationship between estrogen receptor-related breast tumors and lymph node status. Their data contain expressions of 7,219 genes for 49 patients obtained from Affymetrix oligonucleotide arrays. The aim is to predict whether lymph nodes are affected, which is coded as a dichotomous response (for data and Supplementary Information see http://mgm.duke.edu/genome/dna_micro/work/).

We applied PCR, PLS and CFEM to this classification problem, where data were normalized for each gene. Following other analyses of these data, we report the results for misclassification error rates and the selected number of predictor genes based on 3-fold cross-validation, performed for 50 runs. The results for PCR, PLS and CFEM are summarized in Table 5.

3.4 Prognosis of early stage cervical cancer

Biewenga et al. (2008) studied the detection of pelvic lymph node metastases through gene expression profiling. The data consists of 35 patients diagnosed with early stage cervical cancer, of which 16 were found to have lymph node metastases, while the other 19 did not have them, and is accessible through Gene Expression Omnibus (GEO). Biewenga et al. (2008) applied various classifiers (naïve Bayes, decision trees and diagonal linear discriminant analysis) that yielded an average misclassification rate of 35.5% for class prediction. Here, we adopt a similar evaluation procedure as that in Biewenga’s study, conducting random splits of the data, for which the proportion

### Table 3. Misclassification error rates for the test set, comparing PCR, PLS and CFEM for the breast cancer data of van’t Veer et al. (2002)

<table>
<thead>
<tr>
<th>Starting no. of genes $p^*$</th>
<th>50</th>
<th>100</th>
<th>500</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCR</td>
<td>9/19 (6)</td>
<td>5/19 (3)</td>
<td>7/19 (2)</td>
<td>10/19 (3)</td>
</tr>
<tr>
<td>PLS</td>
<td>4/19 (8)</td>
<td>5/19 (1)</td>
<td>6/19 (1)</td>
<td>4/19 (2)</td>
</tr>
<tr>
<td>CFEM</td>
<td>3/19 (5)</td>
<td>3/19 (3)</td>
<td>2/19 (3)</td>
<td>2/19 (2)</td>
</tr>
<tr>
<td>No. of genes selected</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>67</td>
</tr>
</tbody>
</table>

### Table 4. Misclassification error rates for the training set, comparing PCR, PLS and CFEM for the breast cancer data of van’t Veer et al. (2002)

<table>
<thead>
<tr>
<th>Starting no. of genes $p^*$</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCR</td>
<td>38.37% (0.93%, 6)</td>
<td>36.89% (1%, 3)</td>
</tr>
<tr>
<td>PLS</td>
<td>41.52% (0.88%, 8)</td>
<td>34.91% (0.51%, 1)</td>
</tr>
<tr>
<td>CFEM</td>
<td>36.46% (0.61%, 5)</td>
<td>37.15% (0.62%, 3)</td>
</tr>
<tr>
<td>No. of genes selected</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

### Table 5. Misclassification error rates of PCR, PLS and CFEM for the breast tumor data of West et al. (2001)

<table>
<thead>
<tr>
<th>Starting no. of genes $p^*$</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCR</td>
<td>8.99% (1.03%, 2)</td>
<td>8.94% (1.52%, 1)</td>
</tr>
<tr>
<td>PLS</td>
<td>12.82% (1.56%, 1)</td>
<td>11.88% (1.36%, 2)</td>
</tr>
<tr>
<td>CFEM</td>
<td>7.65% (0.83%, 2)</td>
<td>6.71% (0.69%, 1)</td>
</tr>
<tr>
<td>No. of genes selected</td>
<td>16</td>
<td>67</td>
</tr>
</tbody>
</table>
of subjects with and without metastases is kept consistent with the original data (16 to 19) and iterating over 200 runs. In each run, we initialize PCR, PLS and CFEM with the top 500 genes that give rise to the most significant differences between the categories in the training set. For simplicity, we use automatic choices of smoothing (general cross-validation) and truncation parameters (fraction of total variance explained is required to exceed 95%). The best classifiers that yield the smallest misclassification rate in the training set are then applied to the test set for classification evaluation. We used three different sizes for the training set, 10, 20 and 30. The resulting average misclassification rates are listed in Table 6. The performance of CFEM is again quite promising, improving on PCR, PLS and the results reported by Biewenga et al. (2008).

### 3.5 Spectrometric data

To demonstrate the performance of FEM for scrambled spectrometric data, we use a subset of the Tecator data (Borggaard and Thodberg, 1992), available at http://lib.stat.cmu.edu/datasets/tecator. These spectrometric curves are obtained for meat samples and record the absorbance intensity as a function of wavelength (see upper panel of Figure 7, along with the fat content of the sample). These curves are seen to be quite smooth. In order to compare the performance of CFEM with PCR and PLS, we reshuffle the wavelengths across the sample as group 1 and those with fat content above the median as group 2 for both sets. Then the misclassification results for CFEM, PCR, PLS and AR can be compared. For this comparison, the size of training set is varied from 50, 100 to 172. The results are in Table 7, where in the first two scenarios, we report the average misclassification rates obtained from 100 random samples without replacement. From the results in Table 7, we infer that FEM outperforms both PCR and PLS for the three types of training datasets.

### 4 CONCLUSION

In this article, a novel concept regarding classification of high-dimensional data through a FEM approach is proposed, with special emphasis on applications to gene expression microarray data. We introduce a FEM technique (alternatively referred to as Smooth Embedding or ‘SmoothE’) to embed high-dimensional random vectors into a function space. Once the embedding step has been completed, the resulting functional data can be analyzed with various FDA techniques, exploiting smoothness by averaging over the effects of highly correlated gene expressions.

Classification based on FEM works best if combined with predictor variable trimming features, which may include pre-selection of variables and iterating between fitting functional classifiers and variable trimming through thresholding. For classification problems involving high-dimensional predictors, we apply the generalized functional linear model in its special incarnation as functional logistic regression, implemented as functional logistic

---

**Table 6.** Average misclassification error rates of PCR, PLS and CFEM for the cervical cancer data of Biewenga et al. (2008)

<table>
<thead>
<tr>
<th>Size of training set</th>
<th>Error (SE)</th>
<th>10</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCR</td>
<td>36.62% (0.69%)</td>
<td>26.37% (0.65%)</td>
<td>22.50% (1.19%)</td>
<td></td>
</tr>
<tr>
<td>PLS</td>
<td>37.46% (0.74%)</td>
<td>30.03% (0.73%)</td>
<td>24.50% (1.35%)</td>
<td></td>
</tr>
<tr>
<td>CFEM</td>
<td>34.42% (0.62%)</td>
<td>23.93% (0.59%)</td>
<td>17.30% (1.16%)</td>
<td></td>
</tr>
</tbody>
</table>

**Table 7.** Misclassification rate of PCR, PLS, AR and CFEM for the Tecator spectrometric data (Borggaard and Thodberg, 1992)

<table>
<thead>
<tr>
<th>Size of training set</th>
<th>Error (SE, M)</th>
<th>50</th>
<th>100</th>
<th>172</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCR</td>
<td>12.23% (0.38%, 10)</td>
<td>7.23% (0.59%, 4)</td>
<td>4.65% (4)</td>
<td></td>
</tr>
<tr>
<td>PLS</td>
<td>18.70% (0.33%, 10)</td>
<td>18.91% (0.29%, 9)</td>
<td>18.68% (10)</td>
<td></td>
</tr>
<tr>
<td>AR</td>
<td>6.6% (0.4%, 6)</td>
<td>6.33% (0.26%, 10)</td>
<td>5.49% (10)</td>
<td></td>
</tr>
<tr>
<td>CFEM</td>
<td>6.23% (0.4%, 6)</td>
<td>4.28% (0.24%, 7)</td>
<td>2.33% (8)</td>
<td></td>
</tr>
</tbody>
</table>
discriminant analysis. Other classification techniques can be applied to the FPC scores after FEM.

For classifying four landmark microarray datasets and a mass spectrometric data, we obtained excellent performance in terms of both misclassification rates and final number of selected genes, both of which one wishes to minimize in such applications. FEM is promising in comparison with other methods that have been proposed for the classification of high-dimensional microarray data. The method has potential to be more widely useful for the analysis of very high-dimensional data.

**Funding:** National Science Foundation grants (DMS-0806199 and DMS-0505537 in parts to H.-G.M).

**Conflict of Interest:** none declared.

**REFERENCES**


