# A NON PARAMETRIC APPROACH FOR CALIBRATION OF FUNCTIONAL DATA

Noslen Hernández[1], Rolando J. Biscay[2],
Nathalie Villa-Vialaneix[3,4] and Isneri Talavera[1]

[1] *CENATAV, Havana - Cuba*
[2] *Centro de Investigacin en Matemticas, Guanajuato - Mexico*
[3] *SAMM, Université Paris 1 - France*
*and* [4] *INRA, UR875 MIA-T, Castanet Tolosan - France*

**Supplementary Material**

This supplemental material aims at providing further details on the simulation results presented in Section 3 of the article. In particular, several graphics, illustrating every step of the analysis, are included and gives an understandable overview of the estimation on simulated data.

## S1 Detailed analysis of the simulations in Section 3

In this section, we provide further details on the simulation results provided in Section 3 of the article. First recall that the data were simulated in the following way: values for the real random variable $Y$ were drawn from a uniform distribution in the interval $[0, 10]$. $e$ is a Gaussian process independent of $Y$ with zero mean and covariance operator $\Gamma = \sum_{j\geq 1} \frac{1}{j^{(1+0.1)}} v_j \otimes v_j$, where $(v_i)_{i\geq 1}$ is the trigonometric basis of $\mathcal{X} = L_2([0, 1])$ (i.e., $v_{2k-1} = \sqrt{2}\cos(2\pi kt)$, and $v_{2k} = \sqrt{2}\sin(2\pi kt)$). More precisely, for all models, $e$ was simulated by using a truncation of $\Gamma$, $\Gamma(s, t) \simeq \sum_{j=1}^{q} \frac{1}{j^{(1+0.1)}} v_j(t)v_j(s)$ by setting $q = 500$. Then, $X$ was generated by four different models or settings including linear and nonlinear ones.

**M1** a model where $E(X|Y)$ is a linear function of $Y$ expressed on the error eigenfunction basis: $X = Yv_1 + Yv_2 + Yv_5 + Yv_{10} + e$;

**M2** a model where $E(X|Y)$ is a nonlinear function of $Y$ expressed on the error eigenfunction basis: $X = \sin(Y)v_1 + \log(Y + 1)v_5 + e$;

**M3** a model where $E(X|Y)$ is a linear function of $Y$ expressed not on the error eigenfunction basis but on polynomials: $X = Yq_1 + 5Yq_2 + e$, where $q_1 = 2t^3$ and

$q_2 = t^4$. Note that such polynomials have coefficients in the Fourier basis that decay faster than $\frac{1}{j^3}$, and so assumption (A8) is fulfilled;

**M4** a model where $E(X|Y)$ is a nonlinear function of $Y$ expressed using the aforementioned polynomials: $X = \sin(Y)q_1 + 20\log(Y+1)q_2 + e$.

From these 4 models, a training and a test samples with sizes $n_L = 300$ and $n_T = 200$, respectively, were generated. To apply the DBIC method, simulated discretized functions were approximated by continuous functions using a functional basis expansion. Specifically, the discrete data were converted into continuous data (or functional predictors) $X$ by approximation through 128 B-spline basis functions of order 4. Figure 1 shows examples of functional predictors $X$ obtained for models M1 (one of the linear models) and M4 (one of the nonlinear models) for three different values of $y$. The underlying mean functions $r(y) = E(X|Y = y)$ are also given. Depending on $Y$ and on the model, the predictors are more or less noisy but for several cases the level of noise is large and the regression problem should be a hard task. Additional figures (for other simulated models) can be found in Hernández et al. (2010, 2011) in which a short simulation study of the DBIC method is presented.

The DBIC method was used according to the 3 steps described in Section 2. For the first step, the conditional mean $r(y)$ was estimated from the training sample by kernel smoothing (such as in Equation (2.3)). For this, it was necessary to tune the bandwidth parameter $h$. This was done through a 10-fold cross-validation for minimizing the $L_2$-norm between the data and the estimated mean curves in the training sample. That is, the training sample $(x_1, y_1), ..., (x_{n_L}, y_{n_L})$ was randomly partitioned into 10 blocks or folds of approximately the same size, and $h_{opt} = \arg\min_{h\in H} \frac{1}{n_L}\sum_{i=1}^{n_L} \|\hat{x}_i^{(h)} - x_i\|_{L_2}^2$ where $H$ is the search interval for possible values of $h$, and $\hat{x}_i^{(h)}$ is the estimate of the mean $r(y_i)$ using a kernel smoothing with parameter $h$ and the data not belonging to the fold in which $(x_i, y_i)$ is.

Figure 2 shows the results of this step for models M1 (Figure 2, (a)-(f)) and M4 (Figure 2, (g)-(l)). The estimate $\hat{r}(y)(t)$ of the mean is shown both as a function of $t$ (for some $y$ values, the same chosen to show the predictors in Figure 1) and as a function of $y$ (for some $t$ values). Together with the estimate of the mean, the true mean and the data are also plotted. The mean is well estimated in both cases. The linear dependence on $y$ for the case of model M1 (Figure 2, (d)-(f)) and the nonlinear dependence on $y$ for model M4 (Figure 2, (j)-(l)) is well stressed out in these figures.

Once the mean is estimated, the empirical covariance of the residuals and its spectral decomposition is calculated. Figure 3 plots the estimates of eigenvalues *versus* the true eigenvalues, as well as the estimates of the first three eigenfunctions together with the true ones for models M1 (Figure 3, (a)-(d)) and M4 (Figure 3, (e)-(h)). The estimation of eigenvalues and eigenfunctions is also satisfactory in both models. As both models were generated with the same error structure, the spectral decomposition to be estimated was the same for both cases. This corroborates also that the estimated mean, different for both cases, was correctly subtracted giving good estimates of the residuals and consequently good estimates of the spectral decomposition of their covariance

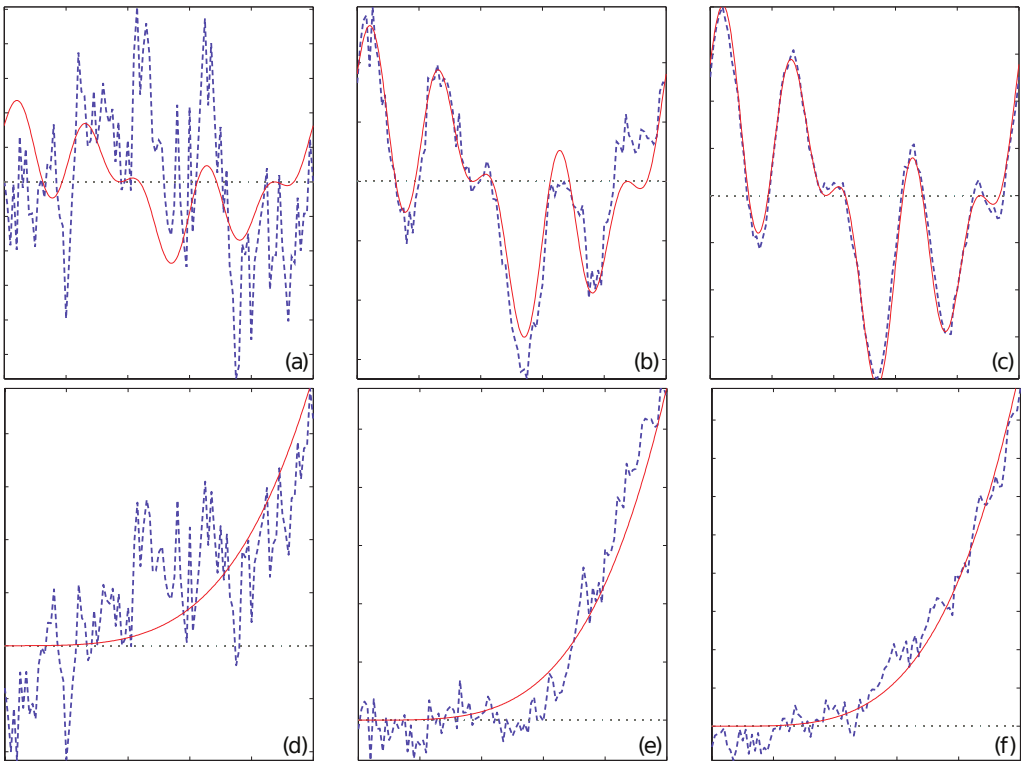


Figure 1: Examples of three functional predictors generated from models M1 (Figures (a)-(c)) and M4 (Figures (d)-(f)). These figures show the predictors $X$ (dashed line) and the mean curves $r(y) = E(X|Y = y)$ (continuous line) as functions of $t$, for the particular values $y = 0.57$ (Figures (a),(d)), $y = 3.20$ (Figures (b),(e)), and $y = 9.83$ (Figures (c),(f)).

operator.

Another hyperparameter involved in the estimation of the regression function $\hat{\gamma}(x)$ is the number $p$ of eigenfunctions (Equation (2.4)) used to estimate $f(x|y)$. Thus, a suitable number $p$ of eigenfunctions has to be chosen. This hyperparameter was selected also by a 10-fold cross-validation for minimizing the root mean squared error (RMSE) criterion on the training sample. Specifically, $p_{opt} = \arg\min_p \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i^{(p)} - y_i\right)^2}$, where $\hat{y}_i^{(p)}$ is DBIC prediction of $y_i$ using the conditional density $\hat{f}^{(p)}$ calculated with $p$ eigenfunctions and the data not belonging to the fold in which $y_i$ is, fold$(i)$. That is, $\hat{y}_i^{(p)} = \frac{\sum_{j\notin \text{fold}(i)} \hat{f}^{(p)}(x_i|y_j)y_j}{\sum_{j\notin \text{fold}(i)} \hat{f}^{(p)}(x_i|y_j)}$.

For model M1 the cross-validation gives the value $p = 15$, which is close to the true one ($p = 10$) according to the model. For model M4 the resulting value was $p = 47$, which

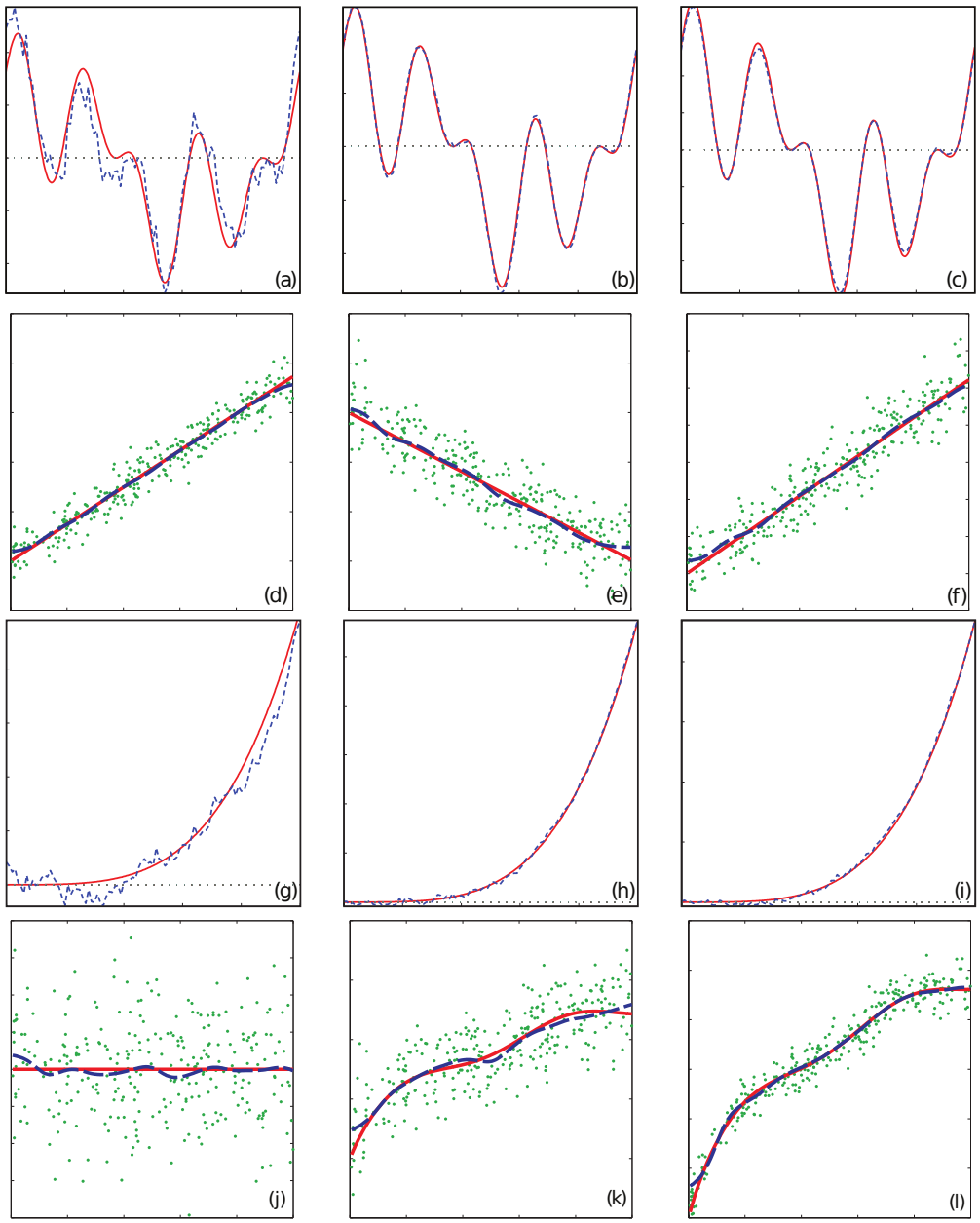


Figure 2: Estimate of the mean $r(y)(t)$ (discontinuous line) as a function of $t$ for various $y$ values (the same as in Figure 1) for models M1 (Figures (a)-(c)) and M4 (Figures (g)-(i)). Estimate of $r(y)(t)$ (dashed line) as function of $y$ for various $t$ values ($t = 0.02, t = 0.71, t = 0.99$) in models M1 (Figures (d)-(f)) and M4 (Figures (j)-(l)). The true mean $r(y)(t)$ (continuous line) and the observed data (points) are also shown.

| Model | DBIC | NWK | DBIC (parametric est. of the mean) |
|---|---|---|---|
| $M1$ | 0.23 | 0.28 | 0.22 |
| $M2$ | 1.71 | 1.91 | X |
| $M3$ | 0.07 | 0.19 | 0.02 |
| $M4$ | 0.35 | 0.47 | X |

Table 1: RMSE achieved by DBIC and NWK for the four simulated models

is larger. Unlike M1, M4 was not built by using the first eigenfunctions of the covariance operator $\Gamma$ in the expression of $E(X|Y)$, hence the need for more eigenfunctions.

Once the estimate $\hat{\gamma}(x)$ is obtained on the basis of the training set, the performance of the DBIC approach was assessed by predicting the $y$ values on the test sample. More precisely, the RMSE was computed: RMSE $= \sqrt{\frac{1}{n_T}\sum_{i=1}^{n_T}(y_i - \hat{y}_i)^2}$, where $y_i$ denotes the observed value of $Y$ in the test sample and $\hat{y}_i$ the corresponding prediction $\hat{\gamma}(x_i)$.

Figure 4 shows the predictions achieved on the test sample by DBIC for models M1 (Figure (a)) and M4 (Figure (c)), which are good in both cases. In order to have a reference to compare with, the standard functional nonparametric kernel estimate (NWK) (Ferraty and Vieu, 2006) was computed from the training sample (using a Gaussian kernel and also tuning the bandwidth parameter by 10-fold cross-validation on the training sample) and its predictions on the test set were calculated. Those predictions are also shown in Figure 4 for model M1 (Figure (b)) and model M4 (Figure (d)).

All these steps were done for each simulated model. Table 1 presents the DBIC RMSE and the NWK RMSE for each of the simulated models. It can be observed that DBIC performs well in all models and outperforms the NWK estimator. The fourth column in the table is the RMSE achieved by DBIC but using a parametric estimation of the mean: instead of estimating the mean using kernel smoothing, the mean was estimated by linear regression (least squares estimates) for models M1 and M3 in which the means are linear functions of $Y$. It can be observed that the RMSE resulting from such a parametric estimates are smaller that those obtained by kernel smoothing. This illustrates that the DBIC approach has the flexibility to incorporate prior knowledge about the mean, if available, and that this additional information can improve the performance.

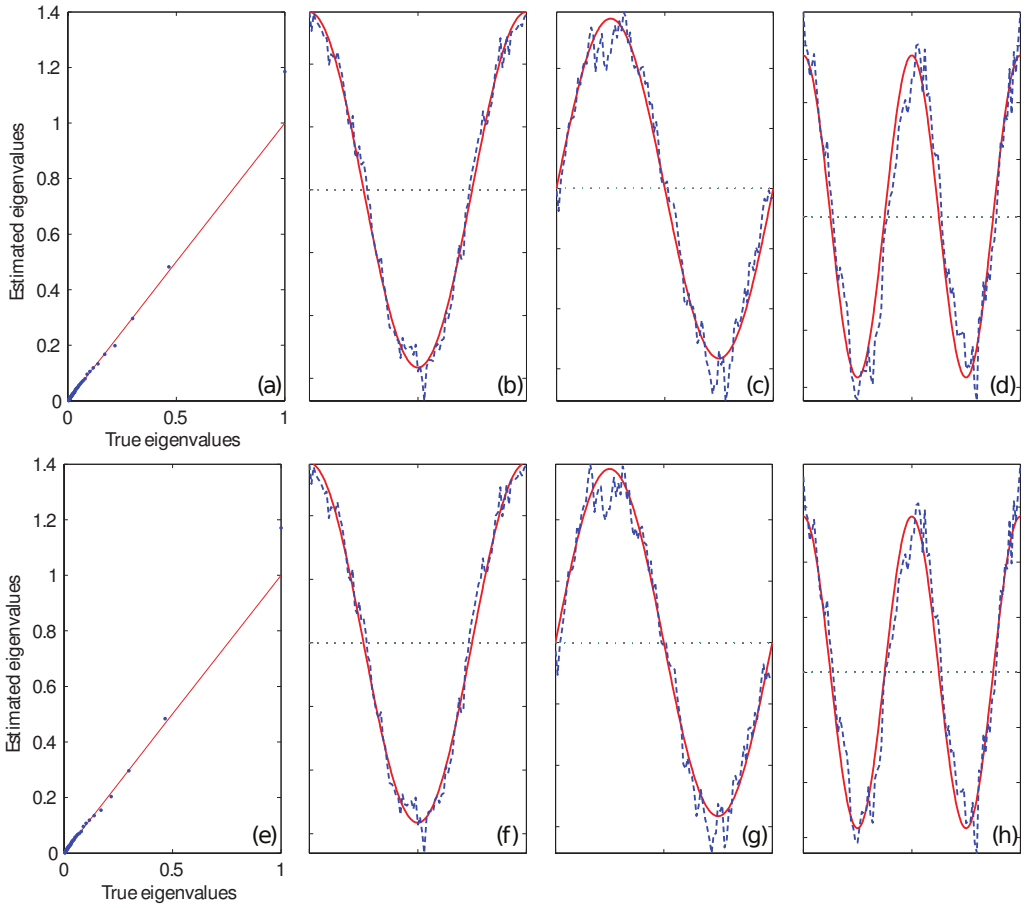


Figure 3: Estimates of the eigenvalues of $e$ versus the true ones for models M1 (Figure a) and M4 (Figure e). Estimates of the first three eigenfunctions of $e$ (dashed lines) and the true first three eigenfunctions of $e$ (continuous lines) in models M1 (Figures (b)-(d)) and M4 (Figures (f)-(h)).

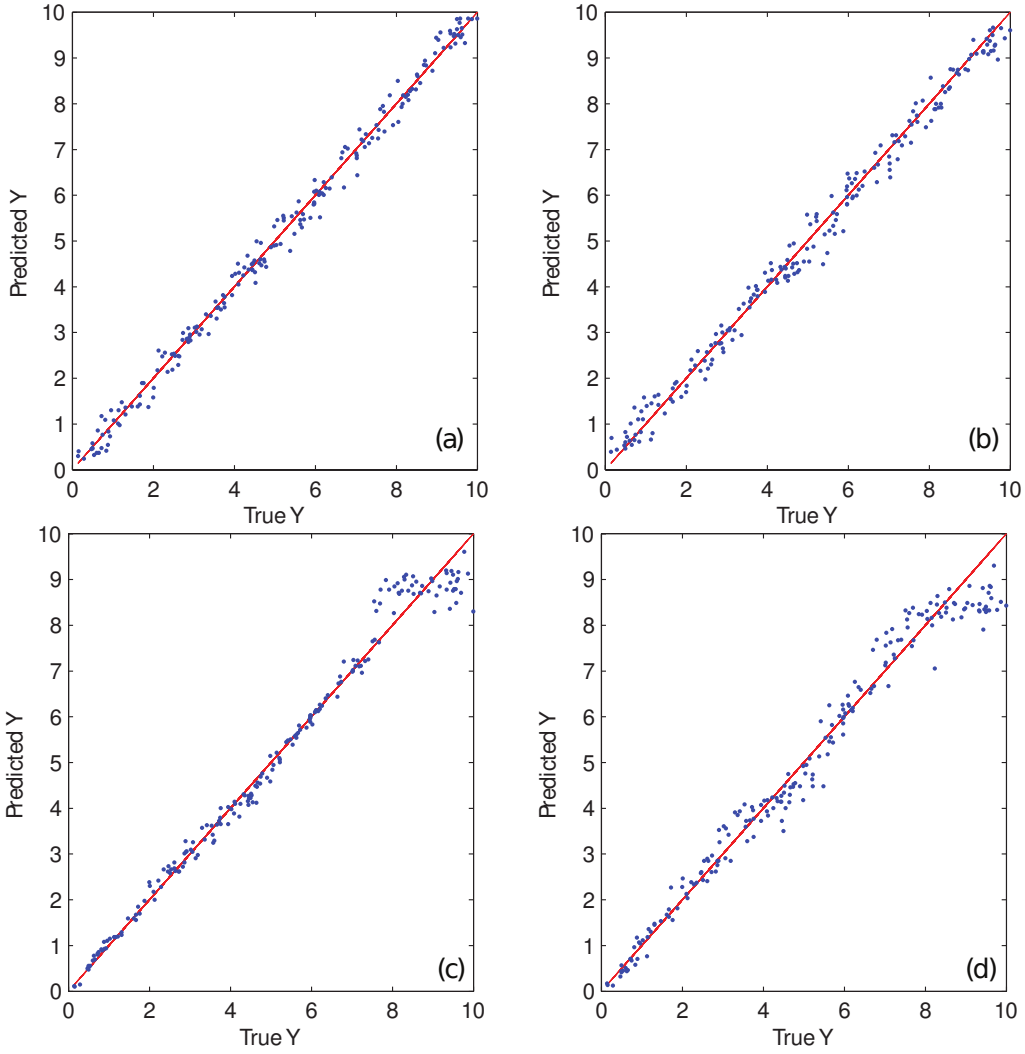


Figure 4: Observed Y values vs. predicted Y values using DBIC for models M1 (Figure (a)) and M4 (Figure (c)), and using NWK for models M1 (Figure (b)) and M4 (Figure (d)).

## References

Ferraty, F. and Vieu, P. (2006). *NonParametric Functional Data Analysis*. Springer.

Hernández, N., Biscay, R., Villa-Vialaneix, N., and Talavera, I. (2011). A simulation study of functional density-based inverse regression. *Revista Investigacion Operacional*, **32**(2), 146–159.

Hernández, N., Biscay, R., Villa-Vialaneix, N., and Talavera-Bustamante, I. (2010). A functional density-based nonparametric approach for statistical calibration. In Bloch, I. and Cesar, R., editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 15th Iberoamerican Congress on Pattern Recognition (CIARP 2010)*, volume 6419 of *Lecture Notes in Computer Science*, pages 450–457, Sao Paulo, Brazil. Springer.