Some Musings on Functional Data

Kathryn Prewitt*

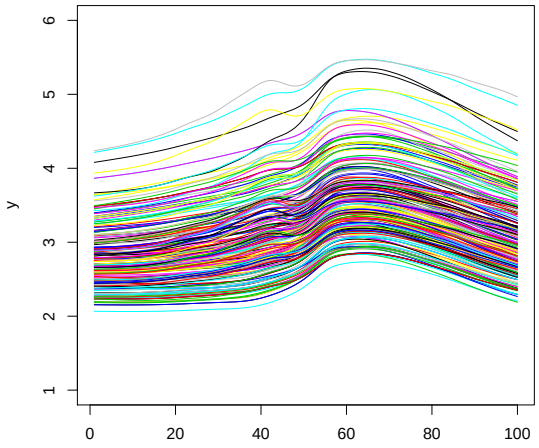
Abstract

Functional data is a current topic of research which was initiated by actual data, i.e. we are collecting a sometimes massive amount of data and the tools to analyze such data are just now catching up. Our main interest for this talk is focused on what we call the NIR (near infrared spectroscopy data set), and we investigate briefly the main methods which are available to analyze such data: PCA (principal component analysis), PLS (partial least squares), kernel smoothing (there are various ways to do this), and SVM (support vector machines).

Key Words: nonparametric functional data analysis, partial least squares, principal component analysis, eigenvectors, eigenfunctions, support vector machine, kernel smoothing

1. The Problem

Functional data is often produced by data which is so dense that we can best describe it as a function. There have been several books written on the subject, see, for example, Ramsey and Silverman (2002) and Ferraty and Vieu (2006). In particular, as we will see, the NIR data is very dense, and when graphed it appears to be a functions.



The horizontal axis in Figure 1 is the wavelength index and the vertical axis is fat absorbance. Unfortunately Figure 1 may or may not show up in this article due to technical difficulties. Often the usual regression methods don't work or are not possible to use because n (the sample size) is smaller smaller than p (the dimension of each sample). The way of handling data such as this was to use only a portion of it, i.e. summarizing the data in some way so as to reduce the dimension. In an effort to use all we have, a variety of functional data methods have appeared.

*Arizona State University School of Mathematical & Statistical Sciences P.O. Box 871804, Tempe, Az, 85287

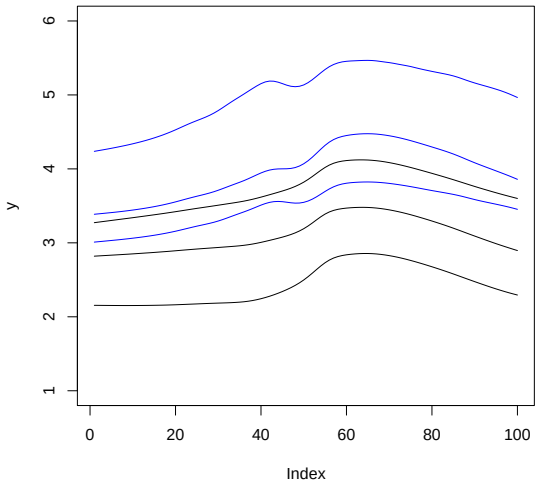


Figure 1: Curves Ordered According to Associated Y Values

2. Notation

We will use the following data notation:

$$\mathbf{X} = \begin{pmatrix} x_1(\nu_1) & \dots & x_1(\nu_{100}) \\ & \ddots & \\ x_{215}(\nu_1) & \dots & x_{215}(\nu_{100}) \end{pmatrix} \qquad \mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_{215} \end{pmatrix}$$

where the experimental units are 215 pieces of finely chopped meat, $n = 215$, $X_i =:$ i^{th} spectrometric curve and $Y_i =:$ fat content of i^{th} piece. We want to predict the fat content based on the function which consists of NIR measurements for the piece of meat. The graphs of the data show that the functions are smooth and highly correlated. However, if we order the data according to the Y values and plot 3 curves associated with high Y values and 3 curves associated with 3 low Y values we do see that the height of the curve wouldn't be a predictor but that at the 750 wavelength the larger Y values have curves with a bump and the lower Y values have curves without the bump.

3. Multiple Regression

We then believe that we know what to expect. Since for this data set, n is larger than p, we could do multiple regression where $n = 215$ and $p = 100$. We regress Y on $\mathbf{X}$ (= design matrix), with the model:

$$Y = \mathbf{X}\beta + \epsilon$$

where $\beta_{p\times 1}$, $\mathbf{X}_{n\times p}, Y_{n\times 1}, \epsilon_{n\times 1}$
The initial results of this analysis shows 16 significant wavelengths

$$(Inter, 1, 2, 28, 29, 45, 54, 55, 60, 63, 64, 68, 69, 73, 79, 80).$$

where the interger values represent the wavelenth index numbered from 1 to 100. However the minimum variance inflation factor is 218717908 and the maximum is 18344003376.

```
 > min(vif(specreg))
[1] 218717908
> max(vif(specreg))
[1] 18344003376
```

The elements of the correlation matrix are all between .96 and .99.

4. PCA

We can consider PCA using principal component regression. We have a design matrix

$$\mathbf{S}_{p\times p} = \frac{1}{n}(\mathbf{X}_{n\times p} - \bar{\mathbf{X}})'(\mathbf{X}_{n\times p} - \bar{\mathbf{X}}) = P\Lambda P'$$

where $\bar{\mathbf{X}}_{n\times p} = (\bar{X}_1^*, \ldots, \bar{X}_p^*)$, $\bar{X}_i^* = \frac{1}{n}\sum_{j=1}^n x_j(\nu_i)$ In addition, we have

- $\Lambda = (\lambda_{ii})_{p\times p}$ where $\lambda_{ii} > \ldots > \lambda_{pp}$
- $P_{p\times p} = (\mathbf{e_1}, \ldots, \mathbf{e_p})$ $P'P = I$
- $(\lambda_{ii}, \mathbf{e_i}) = i^{th}$ (eigenvalue, eigenvector) pair

Our goal is to regress Y on the PC's (principal components).

$$\mathbf{Y} = (\mathbf{X} - \bar{\mathbf{X}})P\beta + \epsilon$$

$$\hat{\beta} = (P'(\mathbf{X} - \bar{\mathbf{X}})'(\mathbf{X} - \bar{\mathbf{X}})P)^{-1}P'(\mathbf{X} - \bar{\mathbf{X}})'\mathbf{Y}$$

1st Prin. Comp (PC)	pth Prin. Comp (PC)
$< \mathbf{e_1}, \mathbf{X} - \bar{\mathbf{X}} >$	$< \mathbf{e_p}, \mathbf{X} - \bar{\mathbf{X}} >$
$\begin{pmatrix} (X_1 - \bar{X})'\mathbf{e_1} \\ \vdots \\ (X_n - \bar{X})'\mathbf{e_1} \end{pmatrix}$	$\begin{pmatrix} (X_1 - \bar{X})'\mathbf{e_p} \\ \vdots \\ (X_n - \bar{X})'\mathbf{e_p} \end{pmatrix}$

We then want to identify the important PC's which are the ones explaining a large percentage of the variation of the design matrix.

Important PC's (?)

	PC1	PC2	PC3	PC4
Standard Deviation	5.111	.488	.280	.174
Proporation of Var	.987	.009	.003	.001
Cumulative Prop.	.987	.996	.999	.999

Standard deviation=$\sqrt{\lambda_{ii}}$; standard dev. of i^{th} PC Scores

We can see that the first 4 PC's account for 99.9% of the variation of the design matrix.

P=$(\mathbf{e_1}, \ldots, \mathbf{e_p})$ is an orthonormal basis for R_p

$$(\mathbf{X} - \bar{\mathbf{X}}) = \sum_{i=1}^{p} < (X_j - \bar{X})', \mathbf{e_i} > \mathbf{e_i}$$

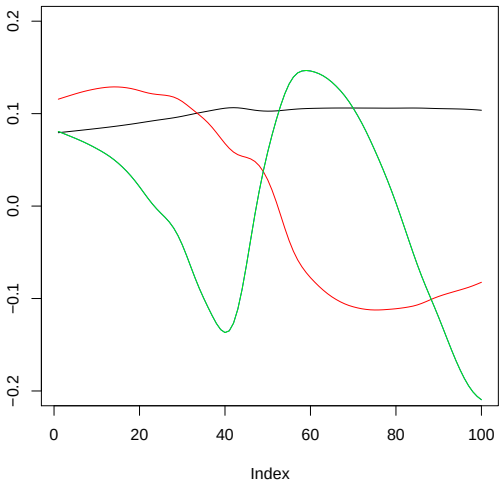
$$(\mathbf{X} - \bar{\mathbf{X}}) \approx \sum_{i=1}^{k} < (X_j - \bar{X})', \mathbf{e_i} > \mathbf{e_i} \; k < p$$

Since 4 PC's account for 99% variation $\Rightarrow k = 4$.

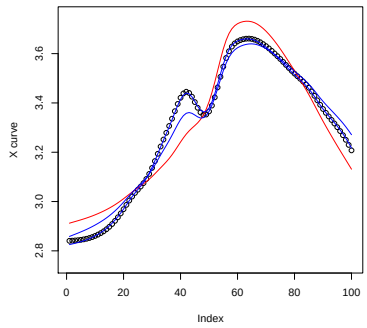
It is of interest to examine the 4 PC's here where

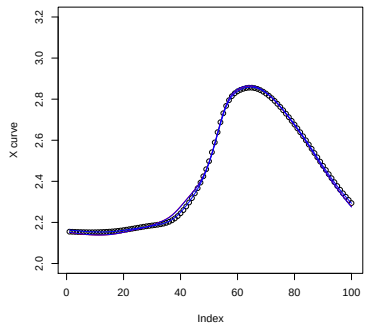
- black=1st, 98.7% variation
- red=2nd, .9% variation
- blue=3rd, .3% variation
- green=4th, .1% variation

While the flat line can said to represent the mean effect, we have difficulty saying what the others really mean and that seems to represent lots of guesswork.

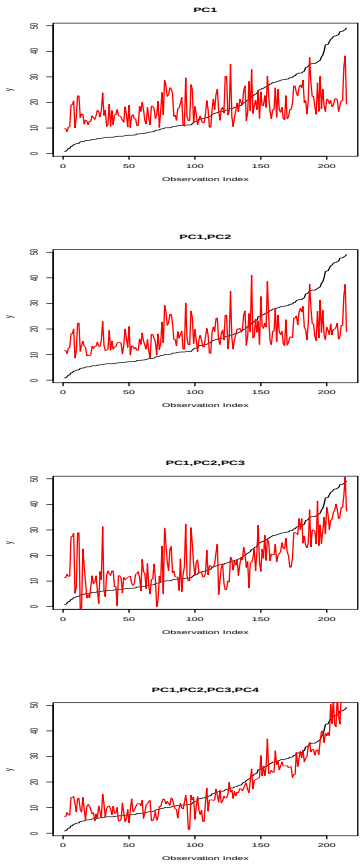


We can take a look at how this method does for X_{215} - curve with $Y_{(215)}$
X_1 - curve with $Y_{(1)}$





We can tell that by appearance, that is, what our eye reports, is that for the curve with the largest Y value and the curve for the smallest Y value seem to be successfully reconstructed with the first 4 PC's. Y vs. $\hat{Y}$ for 4 models[(1).987 (2).996 (3).999 (4).999]



However, in examining the PCR (principal component regression), we find others significant:

PC	St. Deviation
PC14	7.21e-04
PC44	3.57e-05
PC62	1.68e-05
PC58	1.49e-05
PC87	5.57e-06

So, even though the first 4 PC's (in terms of variation explained) are supposed to be important it seems that others are significant and perhaps have some importance too.

SectionNonparametric Principal Component Regression Since thte data was so dense and smooth, we didn't do any smoothing process as we describe below. We provide the model below.

$$\mathbf{Y} = \int (X(t) - \bar{X}(t))\beta(t)\ dt + \epsilon$$

PC Scores in Functional Data:

$$< \mathbf{e_i}, \mathbf{X} - \bar{\mathbf{X}} >= \int \mathbf{e}(\mathbf{u})[\mathbf{X}(\mathbf{u}) - \bar{\mathbf{X}}(\mathbf{u})]\ du$$

Then we have the Karhunen-Loeve dxpansion where eigenfunctions form basis of function space and are obtained from $Cov(X(s), X(t))$ A certain number of the eigenfunctions are selected, often 2. The results in this particular case are similar to PCR. Refer to work by H-G Muller at UC Davis.

There are other functional data methods which can be used by selecting any number of basis function variations. However, it seems there is not a "best" way to select the best basis function. While a certain set of basis functions may explain the variation of the design matrix, there are cases where it doesn't do a good job of predicting Y.

5. Partial Least Squares

While there are several articles about PLS in major statistics journals, there is a huge amount of literature in the chemometrics journals about this topic. PLS is a major competitor for PCA. Sometimes these problems are called calibration problems and proponents of PLS can show examples where the PC with the highest variance explained is actually orthogonal to the Y variable and therefore doesn't have predictive power. The methods seem similar to factor analysis. In particular, with the data in our example, PLS methods are used in industry. There are programs available in R, but it would be hard to compare all the methods because there are so many.

6. Kernel Smoothing - Ferraty, Vieu et al.

A distance between functions is needed so a semi-metric (as they call it) might be what we see below.

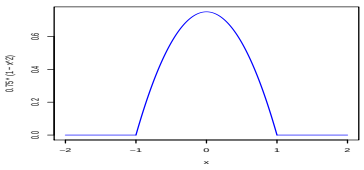
- L_2

$$\sqrt{\int (X_i(t) - X_j(t))^2 \, dt}$$

- 2nd derivative

$$\sqrt{\int (X_i^{(2)}(t) - X_j^{(2)}(t))^2 \, dt}$$

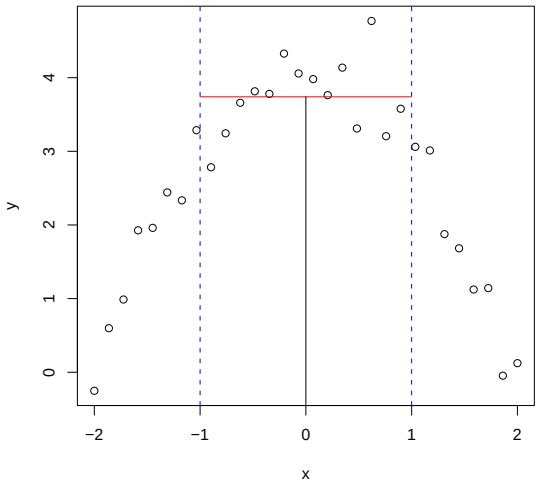
The smoother used by Ferraty et al. is a Nadaraya-Watson type adapted to using a distance measure which is always positive. In the typical Nadaraya-Watson we have the data $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, b=bandwidth and $K(\cdot)$=kernel function (symmetric density)



The estimator is then:

$$\hat{m}(x) = \sum_{i=1}^{n} \frac{K(\frac{x-X_i}{b})Y_i}{\sum K(\frac{x-X_i}{b})} = \sum_{i=1}^{n} c_i Y_i$$

In our problem of interest, we will be estimating when

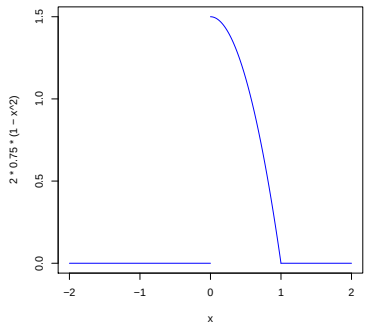


In the above example, the bandwidth (b) is 1, with a window $= [-1, 1]$ A line is fitted within the window using weighted least squares. where the weights

$$= \frac{1}{b} K(\frac{0 - X_i}{b})$$

and the estimator is the height at $x = 0$. So, in general if we wish to estimate the curve at x^*), we follow the procedure outlined above repeatedly. The minimum number of points in the window is 1 but it probably works best with 6.

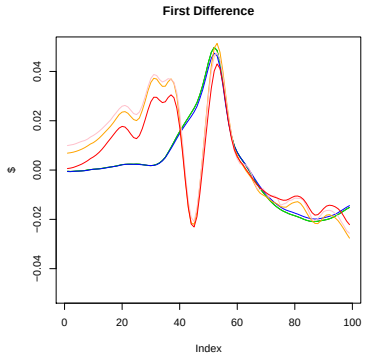
When dealing with functional data where the distance is positive, the kernel's support is postive: $K(\cdot)\ \int_0^\infty K(u)\ du = 1$ e.g.

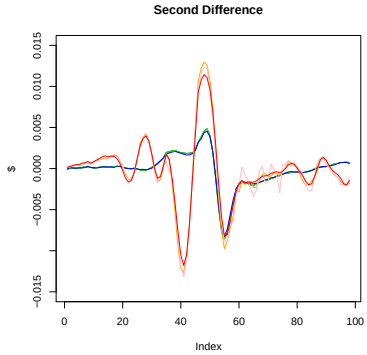


So, instead of the previous setting, we are alwayts estimating on the edge.

- $d(\chi, \chi_\mathbf{i})$ semi-metric distance measure
- L_2 distance for 2nd derivative - used in our example
- $$\hat{r}(\chi) = \sum_{i=1}^{n} \frac{K(\frac{d(\chi,\chi_\mathbf{i})}{b}) Y_i}{\sum K(\frac{d(\chi,\chi_\mathbf{i})}{b})} = \sum_{i=1}^{n} c_i Y_i$$

Black $Y_{(2)}$, Green $Y_{(3)}$, Blue $Y_{(4)}$, Orange $Y_{(213)}$,Pink $Y_{(214)}$, Red $Y_{(215)}$





Mean Square Prediction Error (MSPE) is a measure used in making comparisons between groups. The original sample is split into two subsamples. So, in this data set For example, Ferraty and Vieu used $(X_1, \ldots, X_{160})$ was the learning sample and $(X_{161}, \ldots, X_{215})$: was the testing sample. The measure of performance is

$$MPSE = \frac{1}{55} \sum_{i=161}^{215} (Y_i - \hat{Y}_i)^2 \quad \text{or } \sqrt{MPSE}$$

In a comparison of the four methods with KS being kernel smoothing

	PCR	FPCA	PLS	KS
MPSE	8.2	8.5	7.4	8.2
$\sqrt{MPSE}$	2.87	2.92	2.72	2.8

7. SVM

And finally support vector machine methods. We found a library in R (e1701, svm) which seemed to have numerous choices for parameters. It also seemed that one would need a program to try all sorts of combinations of the parameters because without that care and consideration one might get results such as ours.

```
svmspec1=svm(y1~.,data=specdat1,type="nu",kernel="linear",cost=100
  ,gamma = 1e-04)
pred1=predict(svmspec1,specdat2[,-1])
crossprod(pred1-specdat2[,1])/55
         [,1]
[1,] 668.1261
```

We did find an article by Hernandez et al. (2009) which claimed that three types of SVM had a $.45 \leq \sqrt{MPSE} \leq .78$ and PLS of 1.8. I wasn't able to verify it as I didn't know the details such as which elements of the data were in the training and testing sample.

REFERENCES

Ramsey, Jim and Silverman, B. K. (2002), *Functional Data Analysis* , New York: Springer.

Ferraty, F. and Vieu, P. (2006), *Nonparametric Functional Data Analysis* , New York: Springer.

Hernandez et. al (2009), "Support Vector Regression for Funbctional Data in Multivariate Calibration Problems", *Analytica Chimica Acta*, 110-116.