ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 319-324

Functional Preprocessing for Multilayer Perceptrons

Fabrice Rossi and Brieuc Conan-Guez

Projet AxIS, INRIA, Domaine de Voluceau, Rocquencourt, B.P. 105
78153 Le Chesnay Cedex, France
CEREMADE, UMR CNRS 7534, Université Paris-IX Dauphine,
Place du Maréchal de Lattre de Tassigny, 75016 Paris, France

Abstract. In many applications, high dimensional input data can be considered as sampled functions. We show in this paper how to use this prior knowledge to implement functional preprocessings that allow to consistently reduce the dimension of the data even when they have missing values. Preprocessed functions are then handled by a numerical MLP which approximates the theoretical functional MLP. A successful application to spectrometric data is proposed to illustrate the method.

1 Introduction

Many modern measurement devices are able to produce high resolution data resulting in high dimensional input vectors. An interesting way to handle this type of data is to make explicit use of their internal structure. Indeed, high resolution data can frequently be identified as discretized functions: this is the case for time series (in the time domain as well as in the frequency domain), spectrometric data, weather data (in which we can have both time and location dependencies), etc. Functional Data Analysis (FDA, see [4]) is an extension of traditional data analysis methods to this kind of functional data. In FDA, each individual is characterized by one or more real valued functions, rather than by a vector of $\mathbb{R}^p$. Function estimates are constructed from high dimensional observation vectors and data analysis (in a broad sense) is carried out on those estimates.

A lot of data analysis methods are completely based on simple operations on the considered data: distance, scalar product and linear combination calculations. Those operations can be defined in a satisfactory way in arbitrary Hilbert spaces that include functional spaces (such as L^2). This means that many data analysis methods can be extended to work directly with functional inputs. There are of course some theoretical problems induced by the infinite dimension of the considered spaces, but nevertheless, traditional data analysis methods have been successfully adapted to functional data, both on theoretical and practical point of views. We refer to [4] for a comprehensive introduction to those methods, especially to functional principal component analysis and functional linear models. For regression and discrimination problems, recent developments of FDA include non linear models such as multilayer perceptrons [1, 5] and non parametric models [2, 3].

The most interesting aspect of FDA is the preprocessing phase in which high dimensional input vectors are used to estimate functions. This phase allows to work for instance with missing data, irregularly sampled curves and/or noisy observations. Indeed, as it is not possible to directly manipulate functions, we have to rely on a computer friendly representation: this is obtained thanks to a basis expansion in the functional space, for instance with a B-spline approximation. The choice of a fixed basis allows both to introduce prior knowledge (for instance a Fourier basis for periodic functions such daily temperature observations in a fixed location) and to deal with problems in the data (irregular sampling and noise). Moreover, a regular representation can be used to implement functional transformation such as derivation, integration, etc.

In this paper, we show how a functional preprocessing can become almost mandatory to model efficiently spectrometric observations with missing data.

2 Functional Multilayer Perceptron

In [1, 5], we have proposed an adaptation of Multilayer Perceptrons (MLP) to functional data. This adaptation is based on the fact that MLP neurons operate by computing a scalar product between their input vector and a weight vector. More precisely, a neuron computes a function from a Hilbert space H to $\mathbb{R}$ defined by $N(x) = T(\langle w, x\rangle + b)$, where $\langle ., .\rangle$ denotes the scalar product on H, w denotes the weight vector (which belongs to H like x), b is the real valued threshold and T is an activation function from $\mathbb{R}$ to $\mathbb{R}$.

Numerical MLPs correspond to the particular case where $H = \mathbb{R}^p$ for a given p. Obviously, we can choose $H = L^2(Z)$ to obtain a neuron that works on square integrable functions defined on an input space Z. In this case, $\langle w, x\rangle$ is in fact $\int w(u)x(u)du$ and w is called a weight function. We can also work easily with multiple input functions by using an appropriate Hilbert space. As the output of the proposed neuron is always a real number, the first hidden layer of the MLP is the only one which has to deal with functional inputs. Subsequent layers are traditional numerical neurons. The obtained MLP is called a Functional MLP (FMLP).

We have demonstrated in [5] that the proposed model has the universal approximation property, as long as we aim at approximating continuous functions defined on a compact subset of H.

3 A practical implementation

As explained in the introduction, FDA consists in modeling high dimensional data as functions. We assume therefore that each example is given as a finite set of input/output pairs, i.e. example i corresponds to m^i pairs $(x^i_j, y^i_j)_{1\leq j\leq m^i}$. FDA main assumption is that there is a regular function g^i such that $y^i_j =$

$g^i(x_j^i) + \epsilon_j^i$, where ϵ_j^i is an observation noise. In this model, both m^i, the number of observations, and the $(x_j^i)_{1\leq j\leq m^i}$ can depend on i. This allows to take into account missing data and more generally irregular sampling.

Of course, the model proposed in the previous section cannot be implemented directly to the considered data as this would mean that we have exact knowledge of input functions and that we can calculate exactly integrals. In [1] we proposed an implementation of the FMLP model in which functions are first represented thanks to a basis expansion and then processed by a functional MLP (this is the traditional method in FDA). We showed that both universal approximation and consistent parameter estimation hold for this model.

In [1], we only considered a fixed functional basis. That is, we assume given a Hilbert basis for the functional space, $(\phi_k)_{k\in\mathbb{N}}$ and we approximate g^i thanks to its coordinates on the q first functions of the basis. We have therefore $g^i \simeq \sum_{k=1}^q \alpha_k(g^i)\phi_k$, where α is the projection operator from H to the vectorial space spanned by the q first basis functions. For each g^i, an approximation of the vector $\alpha(g^i)$ is obtained by minimizing the following distortion:

$$\sum_{j=1}^{m^i}\left(\sum_{k=1}^{q}\alpha_k(g^i)\phi_k(x_j^i) - y_i^j\right)^2,$$

which is an approximation of the theoretical distortion $\left\|\sum_{k=1}^q \alpha_k(g^i)\phi_k - g^j\right\|^2$ that defines the projection of g^j on the q first basis functions.

The FMLP neuron is based on scalar product calculation. As g^j is replaced by its orthogonal projection on the q first basis functions, we can restrict ourselves to weight functions in the same vectorial space. We want therefore to calculate $\langle f, g\rangle$ where both f and g belong to $span(\phi_1, \ldots, \phi_q)$. Obviously, we have:

$$\langle f, g\rangle = \sum_{k=1}^{q}\sum_{l=1}^{q}\alpha_k(f)\alpha_l(g)\langle\phi_k, \phi_l\rangle = \sum_{l=1}^{q}\beta_l(f)\alpha_l(g),$$

where $\beta_l(f) = \sum_{k=1}^q \alpha_k(f)\langle\phi_k, \phi_l\rangle$. This equation shows that with an appropriate choice of numerical coefficients, a scalar product between a weight function f and another function g is equal to a linear combination of the coordinates of g. Therefore, the FMLP neuron can be easily implemented as a numerical neuron working on the coordinates of input functions on the chosen truncated basis. The important point is that this way of handling functions is consistent: it is an approximation of the theoretical model which converges to it when the number of evaluation points increases.

4 Functional preprocessing

In practice, it is quite common to end up with a high number of basis functions. Indeed, the standard method used to determine the optimal number of basis functions to retain is to build a cross-validation score for each function for a given truncated basis, and to choose the truncated basis that minimizes the mean cross-validation score for all input functions. For regular functions without noise, we can have almost no reduction in the size of the data. In this situation, FDA is still interesting as it allows to work with irregularly sampled data, but at the expense of dealing with high dimensional input vectors.

The standard way to address the dimension problem is to rely on a functional principal component analysis (PCA) [4]. As its multivariate counterpart, the functional PCA constructs an optimal reduced rank approximation of the original data. In the functional vision, this corresponds to building an optimal truncated basis for the observed functions, that is to choose q orthogonal functions $\psi_1, \dots \psi_q$ such that the following distortion is minimal:

$$\sum_{j=1}^{n} \left\| g^j - \sum_{k=1}^{q} \langle g^j, \psi^k \rangle \psi^k \right\|^2 ,$$

where n denotes the number of functions. When the original functions g^j are regularly sampled and when we have no missing data, the functional PCA can be conducted as a traditional PCA on the sampled curves. When we have more complex data (irregular sampling, missing data, noise, etc.), we first estimate g^j thanks to a standard truncated basis, such a B-spline basis, and the we conduct a functional PCA (which can be in turn obtained thanks to a numerical PCA on preprocessed coordinates).

The interesting point is that the same argument used in the previous section to implement FMLP neuron thanks to coordinates on the truncated basis can be applied to the functional PCA representations. Indeed, the theoretical FMLP calculations can be approximated by a numerical MLP working on the coordinates of the original estimated functions in a basis made of principal functions.

5 An application to spectrometric data

We study in this section spectrometric data from food industry[1]. Each observation is the near infrared absorbance spectrum of a meat sample (finely chopped), recorded on a Tecator Infratec Food and Feed Analyses (we have 215 spectra). More precisely, an observation consists in a 100 channel spectrum of absorbances in the wavelength range 850–1050 nm. The regression problem is to predict the percentage of fat in the meat sample based on the spectrum.

The original contributor of the data reported in [6] very good results: the root mean square prediction error (RMSE) on the test set was 0.42, with an ensemble of 10 MLP (with direct connection between the input and output layer), trained with a weight decay regularization term (three separates weight decays in fact). Meta-parameters of the model (the number of hidden neurons, the weight decays and the number of inputs) were determined by a Bayesian approach. An even more complex procedure with automatic input relevance determination leads to a RMSE of 0.36. An important point is that data are reduced from $\mathbb{R}^{100}$ to $\mathbb{R}^{12}$ by PCA (the number of principal components to keep is determined by the Bayesian approach).

[1] available on statlib: http://lib.stat.cmu.edu/datasets/tecator

In order to investigate the practical interest of functional MLP, we have set up a simpler model: an ensemble of 10 functional MLP, trained with only one weight decay parameter and with no direct connection. Meta-parameters are determined by k-fold cross-validation. We represent functions on a B-spline basis and we conduct a functional PCA. The cross-validation score for function estimation leads to a 61 B-spline basis. As this number is still quite high, the functional PCA is very interesting. The cross-validation selects 14 functional principal components and 3 hidden units. The ensemble functional MLP obtains a RMS of 0.42. This value is not surprising. Indeed the considered functions are very smooth and noiseless (that's why we keep 61 B-splines for 100 evaluation points), and sampling is regular. In this case, the functional PCA gives nearly identical results as the classical PCA conducted on sampled curves.

In order to show the full power of the functional approach, we have removed 10% of the original data. More precisely, we have deleted 10 observations chosen at random out of 100 in each spectrum. The resulting missing data model is the simplest possible one: data are missing at random. With the functional preprocessing, this deletion as no real consequence. Indeed we can still evaluate underlying functions. The cross-validation selects 59 B-splines for the representation of those functions, which is still quite big. Therefore, we perform a functional PCA and submit the coordinates of the original spectra on the functional principal components to the ensemble MLP. The cross-validation selects 12 principal components and 3 hidden neurons, which leads to a RMSE of 0.39. The small improvement is not really meaningful and we can conclude that the deletion of 10 % of the data has no real impact on the prediction quality.

We have compared the functional approach to traditional approaches in which missing data are replaced by estimated valued obtained from other observations. The simplest method consist in replacing missing data by the mean of the corresponding variable. In the current application, this leads to catastrophic performances as the RMSE increases to 8.9.

A more sophisticated method consists in using a k nearest neighbor (k-NN) algorithm: given an input vector in which the p-th coordinate is missing, we calculate its k nearest neighbors among vectors that do not miss this coordinate, and we replace the missing value by the average of the p-th coordinate of the k nearest neighbors. The number of neighbors is determined by cross-validation. This method gives much better results than the mean replacement: with $k = 4$, we obtain a RMSE of 2.0. Nevertheless, this performances are still very bad compared to the functional approach.

Method	Functional	Mean	k-NN	Mean (cs)	k-NN (cs)
RMSE	0.39	8.9	2.0	1.7	0.83

Table 1: RMSE for different missing data reconstruction methods (cs stands for centered and scaled spectra)

The very bad performances of the reconstruction methods are in fact consequences of the true functional nature of the problem. Indeed, it is well known (see e.g. [2, 3]) that chemical properties are related to the shape of the spectrum rather than to its mean value. In order to include this prior knowledge in the reconstruction process, we have used both methods (replacement by the mean or by k-NN) on centered and scaled spectra: each spectrum is first centered with respect to its mean and scaled with respect to its variance. This scaling improves the performances and we obtain a RMSE of 1.7 for the mean replacement and 0.83 for the k-NN replacement (in this case the cross-validation chooses $k = 2$). Even if the preprocessing improves a lot the performances, they are at best twice as bad as the one obtained with the functional approach.

6 Conclusion

The proposed experiment, albeit conducted on real world data, is of course heavily biased toward a functional approach and it is then not very surprising that a functional MLP performs very well on it. We will not therefore claim that the proposed method is a perfect way to handle missing data. Our main claim is that when we are dealing with data that clearly have a functional nature, trying a classical multivariate analysis is not a good idea. If the data are simple, i.e. noise free, regularly sampled and with no missing observations, FDA approaches give similar results to multivariate methods. But when we have complicated data, a functional preprocessing allows to use efficiently the prior knowledge that data are functions and performs therefore much better than multivariate approaches. Moreover, theoretical results give to the functional approaches a sound mathematical framework, exactly as for finite dimensional methods.

References

[1] Brieuc Conan-Guez and Fabrice Rossi. Multilayer perceptrons for functional data analysis: a projection based approach. In José R. Dorronsoro, editor, *Artificial Neural Networks – ICANN 2002*, pages 667–672, Madrid, August 2002. Springer.

[2] Frédéric Ferraty and Philippe Vieu. The functional nonparametric model and application to spectrometric data. *Computational Statistics*, 17(4), 2002.

[3] Frédéric Ferraty and Philippe Vieu. Curves discriminations: a nonparametric functional approach. *Computational Statistics and Data Analysis*, 44(1–2):161–173, 2003.

[4] Jim Ramsay and Bernard Silverman. *Functional Data Analysis.* Springer Series in Statistics. Springer Verlag, June 1997.

[5] Fabrice Rossi, Brieuc Conan-Guez, and François Fleuret. Theoretical properties of functional multilayer perceptrons. In *Proceedings of ESANN 2002*, pages 7–12, Bruges, Belgium, April 2002.

[6] Hans Henrik Thodberg. A review of Bayesian neural networks with an application to near infrared spectroscopy. *IEEE Trans. on Neural Networks*, 7(1):56–72, 1996.