Gaussian Fitting Based FDA for Chemometrics

Tuomas Kärnä and Amaury Lendasse

Helsinki University of Technology, Laboratory of Computer and Information Science,
P.O. Box 5400 FI-02015, Finland
tuomas.karna@hut.fi, lendasse@hut.fi

Abstract. In Functional Data Analysis (FDA) multivariate data are considered as sampled functions. We propose a non-supervised method for finding a good function basis that is built on the data set. The basis consists of a set of Gaussian kernels that are optimized for an accurate fitting. The proposed methodology is experimented with two spectrometric data sets. The obtained weights are further scaled using a Delta Test (DT) to improve the prediction performance. Least Squares Support Vector Machine (LS-SVM) model is used for estimation.

1 Introduction

In Functional Data Analysis [1] samples are treated as functions instead of traditional discrete vectors. A crucial part of FDA is the choice of basis that allow the functional representation. Commonly used bases are b-splines, Fourier series or wavelets. However, it is appealing to build a problem-specific basis that employs the statistical properties of the data at hand.

In literature, there are examples of finding the optimal set of basis functions that minimize the fitting error, such as Functional Principal Component Analysis [1]. The basis functions obtained by Functional PCA usually have global support (i.e. they are non-zero throughout the data interval). Thus these functions are not good for encoding spatial information of the data. The spatial information, however, may play a major role in many fields, such as spectroscopy. For example, often the measured spectra contains spikes or ditches at certain wavelengths that correspond to certain substances in the sample. Therefore these areas are bound to be relevant for estimating the quantity of these substances.

We propose that locally supported functions, such as Gaussian kernels, can be used to encode this sort of spatial information. In addition, variable selection can be used to select the relevant kernels from the irrelevant ones. Selecting important variables directly on the raw data is often difficult due to high dimensionality of data; computational cost of variable selection methods (such as Backward-Forward [5]) grows exponentially with the number of variables. Therefore, wisely placed Gaussian kernels are proposed as a tool for encoding spatial information while reducing data dimensionality so that other more powerful information processing tools become feasible. Delta Test (DT) based scaling of variables is suggested for improving the prediction performance.

F. Sandoval et al. (Eds.): IWANN 2007, LNCS 4507, pp. 186–193, 2007.

The methodology is presented in Section 2 starting with a brief overview. The optimization of Gaussian kernels is explained in Section 2.2, DT based scaling in Section 2.3 and LS-SVM model in Section 2.4. Section 3 presents two real world applications with results. Finally the concluding remarks are drawn in Section 4.

2 Methodology

Consider a problem where the goal is to estimate a certain quantity $p \in \mathbb{R}$ from a measured spectrum X based on the set of N training examples $C_L = (X_j, p_j)_{j=1}^N$. In practice, the spectrum X_j is a set of discretized measurements $(x_i^j, y_i^j)_{i=1}^m$ where $x_i^j \in \mathbb{R}$ stand for the wavelength and $y_i^j \in \mathbb{R}$ is the response.

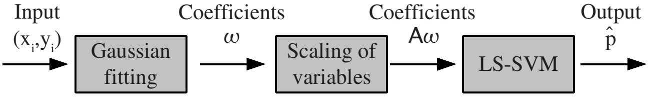

Fig. 1. Outline of the prediction method

Adopting the FDA framework [1], our goal is to build a prediction model F so that $\hat{p} = F(X)$. The argument X is a real-world spectrum, i.e. a continuous function that maps wavelengths to responses. Without much loss of generality it can be assumed that $X \in L_2([a,b])$. However, since the spectrum X is unknown and infinite dimensional it is approximated with a q dimensional vector $\boldsymbol{\omega} = \mathcal{P}(X)$, $\mathcal{P} : L_2 \longmapsto \mathbb{R}^q$. In this case our prediction model can be reformulated as $\hat{p} = F(\boldsymbol{\omega})$.

Figure 1 presents a graph of the overall prediction method. Gaussian fitting is used for the approximation of X. The obtained vectors $\boldsymbol{\omega}$ are further scaled by a diagonal matrix A before the final LS-SVM modeling. The following sections explain these steps in greater detail.

2.1 Finite Dimensional Representation of *X*

Because the space $L_2([a,b])$ is infinite dimensional, it is necessary to consider some finite dimensional subspace $\mathcal{A} \subset L_2([a,b])$. We define $\mathcal{A}$ by a set of Gaussian kernels

$$\varphi_k(x) = e^{-\frac{\|x-t_k\|^2}{\sigma_k^2}}, \; k = 1, \ldots, q \tag{1}$$

where t_k is the center and σ_k is the width parameter. If the kernels are linearly independent, the set $\varphi_k(x)$ spans a q dimensional normed vector space and we can write $\mathcal{A} = \mathtt{span}\{\varphi_k\}$. A natural choice for the norm is the L_2 norm: $\| \hat{f} \|_{\mathcal{A}} = \left(\int_a^b \hat{f}(x)^2 dx \right)^{1/2}$.

Now X can be approximated using the basis representation $\hat{X}(x) = \boldsymbol{\omega}^T \boldsymbol{\varphi}(x)$, where $\boldsymbol{\varphi}(x) = [\varphi_1(x), \varphi_2(x), \ldots, \varphi_q(x)]^T$. The weights $\boldsymbol{\omega}$ are chosen to minimize the square error:

$$\min_{\boldsymbol{\omega}} \sum_{i=1}^{m} \mid y_i - \boldsymbol{\omega}^T \boldsymbol{\varphi}(x_i) \mid^2 \tag{2}$$

Using this basis, any function $\hat{X} \in \mathcal{A}$ is uniquely determined by the weight vector $\boldsymbol{\omega}$. This suggests that, under certain constraints, it is equivalent to analyze the discrete weight vectors instead of the continuous functions $\hat{X}$.

Radial symmetric models (such as the LS-SVM presented in Section 2.4) depend only on the distance metric $d(\cdot,\cdot)$ in the input space. Thus, we require that the mapping from $\mathcal{A}$ to $\mathbb{R}^q$ is isometric, i.e. $d_{\mathcal{A}}(\hat{f},\hat{g}) = d_q(\boldsymbol{\alpha},\boldsymbol{\beta})$ for any functions $\hat{f}(x) = \boldsymbol{\alpha}^T\boldsymbol{\varphi}(x)$ and $\hat{g}(x) = \boldsymbol{\beta}^T\boldsymbol{\varphi}(x)$. The first distance is calculated in the function space and the latter one in $\mathbb{R}^q$. In the space $\mathcal{A}$, distances are defined by the norm $d(\hat{f},\hat{g}) = \parallel \hat{f} - \hat{g} \parallel_{\mathcal{A}}$. Now a simple calculation gives

$$\parallel \hat{f} - \hat{g} \parallel_{\mathcal{A}}^2 = \int_a^b (\sum_{k=1}^q (\alpha_k - \beta_k)\varphi_k(x))^2 dx = (\boldsymbol{\alpha} - \boldsymbol{\beta})^T \varPhi (\boldsymbol{\alpha} - \boldsymbol{\beta})$$

where $\varPhi_{i,j} = \int_a^b \varphi_i(x)\varphi_j(x)dx$. This implies that if the basis is orthonormal, the matrix $\varPhi$ becomes an identity matrix and the distances become equal, i.e. $\parallel \hat{f} - \hat{g} \parallel_{\mathcal{A}} = \parallel \boldsymbol{\alpha} - \boldsymbol{\beta} \parallel_2 = ((\boldsymbol{\alpha} - \boldsymbol{\beta})^T(\boldsymbol{\alpha} - \boldsymbol{\beta}))^{1/2}$. Unfortunately this is not the case with Gaussian kernels and a linear transformation $\tilde{\boldsymbol{\omega}} = \mathbf{U}\boldsymbol{\omega}$ is applied. Here the matrix $\mathbf{U}$ is the Cholesky decomposition of $\boldsymbol{\varPhi} = \mathbf{U}^T\mathbf{U}$. In fact, the transformed weights $\tilde{\boldsymbol{\omega}}$ are related to a set of new orthonormal basis functions $\tilde{\boldsymbol{\varphi}} = \mathbf{U}^{-T}\boldsymbol{\varphi}$.

2.2 Finding an Optimal Gaussian Basis

When the locations and widths of the Gaussian kernels are known, the weights $\boldsymbol{\omega}$ are obtained easily by solving the problem (2). The solution is the well-known pseudoinverse $\boldsymbol{\omega} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{y}$ [4], where $\mathbf{y} = [y_1, y_2, \ldots, y_m]^T$ are the values to be fitted and $\mathbf{G}_{i,j} = \varphi_j(x_i)$.

Since the kernels are analytical functions, the locations and widths can be optimized for a better fit. The average fitting error of all functions is obtained by averaging Eq. (2) over all of the sample inputs $j = 1, \ldots, N$. Using the matrix notation given above, it can be formulated as

$$E = \frac{1}{2N} \sum_{j=1}^N \left(\mathbf{G}\boldsymbol{\omega}_j - \mathbf{y}_j\right)^T \left(\mathbf{G}\boldsymbol{\omega}_j - \mathbf{y}_j\right).$$

The partial derivates are

$$\frac{\partial E}{\partial t_k} = \frac{1}{N} \sum_{j=1}^N \left(\mathbf{G}\boldsymbol{\omega}_j - \mathbf{y}_j\right)^T \mathbf{G}_k^{(t)} \omega_{j,k}$$

$$\frac{\partial E}{\partial \sigma_k} = \frac{1}{N} \sum_{j=1}^N \left(\mathbf{G}\boldsymbol{\omega}_j - \mathbf{y}_j\right)^T \mathbf{G}_k^{(\sigma)} \omega_{j,k},$$

where the notation $\mathbf{G}_k^{(t)}$ and $\mathbf{G}_k^{(\sigma)}$ stand for the k-th column of $\mathbf{G}$ differentiated with respect to t_k and σ_k, respectively.

Knowing the gradient, the locations and the widths are optimized using unconstrained non-linear optimization. Actually, the problem is constrained to $\sigma > 0$ but the kernel (1) is an even function with respect to σ and the constraint can be relaxed. In this paper, Broyden-Fletcher-Goldfarb-Shanno (BFGS) Quasi-Newton method with line search is used. The formulation of the BFGS algorithm is out of the scope of this paper. The reader can refer to [2], for example.

2.3 Scaling

Variable scaling can be seen as a generalization of variable selection; instead of restricting the scalars to attain either values 0 or 1, the entire range $[0, 1]$ is allowed. In this section, we present a method for choosing the scaling using Delta Test (DT).

The scalars are optimized by iterative Forward-Backward Selection (FBS) (see [5], for example). FBS is usually used for variable selection, but it can be extended to scaling as well; Instead of turning scalars from 0 to 1 or vice versa, increases by $1/h$ (in the case of forward selection) or decreases by $1/h$ (in the case of backward selection) are allowed. Integer h is a constant grid parameter.

DT is a method for estimating the variance of the noise within a data set. Having a set of general input-output pairs $(\mathbf{x}_i, y_i) \in \mathbb{R}^m \times \mathbb{R}$ and denoting the nearest neighbor of $\mathbf{x}_i$ by $\mathbf{x}_{NN(i)}$, the variance estimate is

$$\delta = \frac{1}{2N} \sum_{i=1}^{N} \left| y_{NN(i)} - y_i \right|^2 ,$$

where $y_{NN(i)}$ is the output of $\mathbf{x}_{NN(i)}$. DT is useful in evaluation of correlation of random variables and therefore it can be used for scaling: The set of scalars that give the smallest δ is selected.

2.4 LS-SVM

LS-SVM is a least square modification of the Support Vector Machine (SVM) [3]. The quadratic optimization problem of SVM is simplified so that it reduces into a linear set of equations. Moreover, regression SVM usually involves three unknown parameters while LS-SVM has only two; the regularization parameter γ and the kernel width θ.

Consider a set of N training examples $(\mathbf{x}_i, y_i)_{i=1}^{N} \in \mathbb{R}^m \times \mathbb{R}$. The LS-SVM model is $\hat{y} = \mathbf{w}^T \boldsymbol{\psi}(\mathbf{x}) + b$, where $\boldsymbol{\psi} : \mathbb{R}^m \longmapsto \mathbb{R}^n$ is a mapping from the input space onto a higher dimensional hidden space, $\mathbf{w} \in \mathbb{R}^n$ is a weight vector and b is a bias term. The optimization problem is formulated as

$$\min_{\mathbf{w},b} \; J(\mathbf{w}, e) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\gamma \sum_{i=1}^{N} e_i^2,$$
$$\text{so that} \quad y_i = \mathbf{w}^T \boldsymbol{\psi}(\mathbf{x}_i) + b + e_i,$$

where e_i is the prediction error and $\gamma \geq 0$ is a regularization parameter. The dual problem is derived using Lagrangian multipliers which leads into a linear KKT system that is easy to solve [3]. Using the dual solution, the original model can be reformatted as

$$\hat{y} = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b,$$

where the kernel $K(\mathbf{x}, \mathbf{x_i}) = \boldsymbol{\psi}(\mathbf{x})^T \boldsymbol{\psi}(\mathbf{x}_i)$ is a continuous and symmetric mapping from $\mathbb{R}^m \times \mathbb{R}^m$ to $\mathbb{R}$ and α_i are the Lagrange multipliers. It should be emphasized that although we formally define the high dimensional hidden space $\mathbb{R}^n$ and the mapping $\boldsymbol{\psi}(\mathbf{x})$, there is no need to compute anything in the hidden space; The knowledge of the kernel K is enough. A widely-used choice for is the standard Gaussian kernel $K(\mathbf{x}_1, \mathbf{x}_2) = \mathtt{exp}\{-\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2/\theta^2\}$.

3 Application

3.1 Data Sets

The proposed prediction method was tested on two spectrometric data sets from the food industry. Tecator data set consists of absorption spectra and fat contents of 215 samples of minced pork meat [6]. Each spectrum has 100 values corresponding to wavelengths from 850nm to 1050nm. The accuracy of the measured fat content is 1 per cent. The first 172 spectra were used as a learning set C_L and the remaining 43 were used as a test set C_T. The training set is illustrated in Figure 2.

The second data set contains 124 measured Near Infrared (NIR) absorption spectra of wine samples and the goal is to determine the percentage of alcohol. Each spectrum has 256 variables corresponding to wavenumbers from 400 to $4000cm^{-1}$ [5]. Alcohol content ranges from 7.48 per cent to 18.5 per cent and

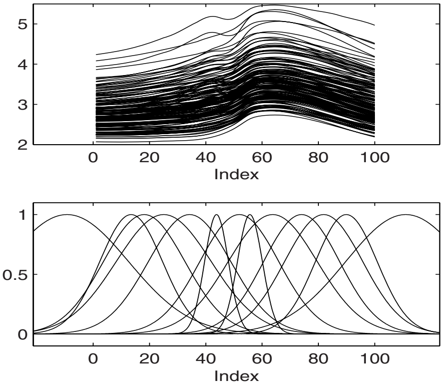

Fig. 2. Tecator dataset and 13 optimized kernels

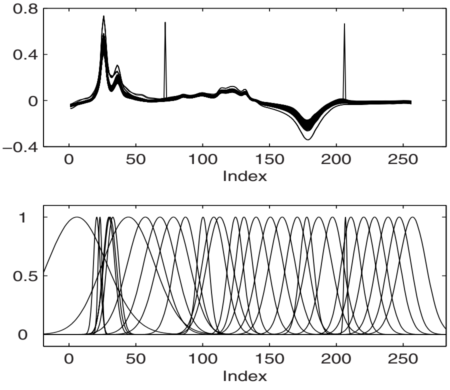

Fig. 3. Wine dataset and 30 optimized kernels

the accuracy is three digits. First 94 spectra were used as a learning set C_L while the remaining 30 were regarded as a test set C_T. The spectra are illustrated in Figure 3.

3.2 Training

The Gaussian basis was optimized in the learning set C_L as described in Section 2.2. Number of kernels ranged from 3 to 35 and initially the kernels were equally distributed. The obtained weights $\boldsymbol{\omega}$ were transformed using the Cholesky matrix. Next, the DT scaling method was applied to improve accuracy. For a reference non-scaled weights were also experimented.

Finally, a LS-SVM model was trained using a two-dimensional grid search and 10-fold cross validation in C_L. To obtain reliable values for γ and θ, a 10-by-10 grid was computed four times starting from a coarse global grid and moving on to a finer one near the optimum. The error measure was normalized mean square error NMSE_L (averaged over the 10 cross validation results). To evaluate the generalization performance the best model was simulated in the test set C_T and NMSE_T was computed,

$$\text{NMSE}_T = \Big(\frac{1}{N_T}\sum_{j\in C_T}(p_j - \hat{p}_j)^2\Big)\Big/\Big(\frac{1}{N_L + N_T}\sum_{j\in C_L\cup C_T}(p_j - \bar{p})^2\Big).$$

3.3 Results and Discussion

TECATOR Data Set. The obtained results are presented in Table 1. The best results were obtained using 13 kernels. The best basis is plotted in Figure 2. It can be seen that there are wide functions at the both ends of the spectrum where the data is smoother and two narrow kernels near the center where there is more variation in the data.

Scaling improves the NMSE by 20 per cent. Ten variables are assigned a nonzero scalar and the corresponding ten orthonormal basis functions are plotted in Figure 4. Although the functions cover the whole spectrum, the most important ones (i.e. ones with high scalar) are related to frequencies in the center.

Comparing to other results in the literature, the performance is very good, although not the best. Thodberg reports a RMSE (calling it Standard Error of Prediction, SEP) 0.36 obtained with a committee of Bayesian neural networks [6] and Vila et. al. report even better RMSE (0.34) for another Bayesian neural network method [7]. The RMSE of our method is 0.43 (LS-SVM with scaling), which is better than the results reported in [8], [10] and [9].

Wine Data Set. In the case of the wine data set, 30 kernels (plotted in Figure 3) were selected. The locations of the narrow kernels coincide to the spikes in the data. Especially, there are many narrow kernels between indexes 20 to 40.

Scaling improves the performance by 15 per cent. Interestingly enough, only four variables obtain a non-zero scalar. This implies that the majority of the

Table 1. Results (NMSE_T) for the Tecator data set and the wine data set

Tecator		Wine	
LS-SVM	0.00148	LS-SVM	0.01004
LS-SVM + Scaling	**0.00116**	LS-SVM + Scaling	**0.00849**

data is irrelevant to the prediction and can be discarded. The selected functions are presented in Figure 5. It can be seen that the first three variables are related to the indexes from 20 to 40. Thus one can conclude that this area is highly correlated to the alcohol content.

Comparing to literature, Benoujdit et al. have reported a NMSE 0.0009 using a Radial Basis Function Network with FBS on the raw data itself [5]. They selected only 20 variables among the 256 which further stresses the fact that most of the variables are irrelevant.

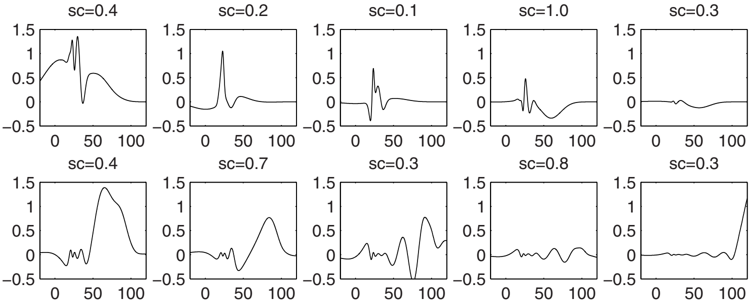

Fig. 4. Tecator data set: selected orthogonal basis functions

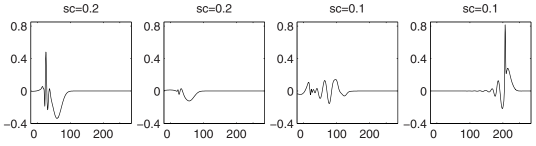

Fig. 5. Wine data set: selected orthogonal basis functions

4 Conclusions

This paper deals with the problem of finding a good set of basis functions for dimension reduction. We have proposed a Gaussian kernel based method where the kernels are optimized for an accurate fit. When combined with an LS-SVM model, our results verify that the basis indeed follows the nature of the original data. And what is more, the basis is useful in the determination of analytical variables from spectral data. The Delta Test based scaling further improves the prediction performance and provides a tool for interpreting the importance of the inputs.

In literature Bayesian networks have been reported to perform slightly better. Thus the authors believe that the proposed method could be improved by replacing the LS-SVM model by a Bayesian network. The fact that the basis is optimized for an accurate fit instead of prediction performance is visible in the wine data set: direct variable selection has been reported to yield better results [5]. However, it is much more time consuming and, on the other hand, one should notice that the obtained errors are already smaller than the numerical accuracy of the original data. Therefore we can conclude that the proposed Gaussian fitting provides a fast tool for dimension reduction.

Acknowledgments

The authors are grateful to Prof. Marc Meurens for providing the wine data set.

References

1. Ramsay, J., Silverman, B.: Functional Data Analysis Springer Series in Statistics. Springer, Heidelberg (1997)
2. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming, Theory and Algorithms. John Wiley and Sons, New York (1993)
3. Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific Publishing Co., Singapore (2002)
4. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn., Prentice Hall Inc., New York (1999)
5. Benoudjit, N., Cools, E., Meurens, M., Verleysen, M.: Chemometric calibration of infrared spectrometers: selection and validation of variables by non-linear models. Chemometrics and Intelligent Laboratory Systems 70, 47–53 (2004)
6. Thodberg, H.: A Review of Bayesian Neural Networks with an Application to Near Infrared Spectroscopy. IEEE Transactions on Neural Networks 7, 56–72 (1996)
7. Vila, J., Wagner, V., Neveu, P.: Pascal Neveu: Bayesian Nonlinear Model Selection and Neural Networks: A Conjugate Prior Approach. IEEE Transactions on Neural Networks 11, 265–278 (2000)
8. Rossi, F., Lendasse, A., François, D., Wertz, V., Verleysen, M.: Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. Chemometrics and Intelligent Laboratory Systems 80, 215–226 (2006)
9. Aneiros-Pérez, G., Vieu, P.: Semi-functional partial linear regression. Statistics and Probability Letters 76, 1102–1110 (2006)
10. Ferré, L., Yao, A.: Smoothed Functional Inverse Regression. Statistica Sinica 15, 665–683 (2005)