ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 15, No. 2, 2020, *pp*.134-140

Classification with application to Functional Data based on Gaussian process

Xin Liu, Chunzheng Cao
School of Mathematics and Statistics, Nanjing University of Information Science & Technology,
Nanjing 210044, China

(Received June 01, 2020, accepted July 05, 2020)

Abstract. In this paper, we briefly introduce four methods for functional classification. To compare the effects of the four models, we generate the data from Gaussian process based on a functional mixed-effects model, square exponential kernel is used in random-effect term to describe the nonlinear structure of the data. The outcomes show that the two functional classification models have a better prediction correct rate than the two machine learning classification models.

Keywords: functional classification, functional mixed-effects model, kernel function.

1. Introduction

Functional data analysis (FDA) is used in many research fields which is of great theoretical and practical value, for example: spatio-temporal data analysis in meteorology [1], pharmacokinetic analysis in medicine [2], genetic profiling analysis in biology [3, 4], image data analysis which is ultra-high dimension [5] and the clustering and prediction of traffic flow data [6]. For functional data, there is a strong correlation between variables, there are also complex correlations between the observations of the variables for each subject, these do not meet the assumptions of the common statistical analysis methods. And FDA can solve these questions.

An important concern of functional data analysis is classification, which means that we want to assign an individual to a pre-designed discrete category based on the observed functional or image data. The existing functional data classification methods can be divided into the following three categories: (1) Based on probability density [7]. (2) Based on the algorithm [8]. (3) Based on regression [9]. Although there are many literatures discussing the classification of functional data, there are still many problems that have not been considered: the influence of covariates on classification cannot be considered, the prediction after classification and the correlation between each subject is not taken into account. In this paper, we will propose a model which can solve these problems. There are also some classical machine learning classification method such as BP neural network [10] and SVM [11], which are very mature approaches.

The rest of this paper is organized as follows. In section 2, we briefly introduce four classification models: two functional classification models and two machine learning classification models and then give their Parameter estimation process. In section 3, we design a simulation study for the models proposed in this paper and compare the performance between the four models. Finally, a brief summary was given in section 4.

2. Model

In this section, we will briefly introduce four classification model: two functional classification models and two machine learning classification models.

2.1. Functional Generalized Linear Model

For Generalized linear model (GLM), we have a general structure

$$g(\mu) = \beta_0 + \beta_1^T \boldsymbol{X}, \tag{1}$$

where $\boldsymbol{X}$ is predictor, $\mu = E(Y;\theta,\phi)$, Y is response variable with density

$$p(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}, \tag{2}$$

and $g(\cdot)$ is a link function. But when the $\boldsymbol{X}$ is functional ,i.e.$\boldsymbol{X}(t)$, the GLM model may not be used, what we will do is to replace the summation over the dimensional space with an integral over the infinite dimensional one,

$$g(\mu) = \beta_0 + \int \omega_1(t)X(t)dt$$

although t is infinite dimensional in theory, in practice t is a finite set of time points. Now we use some basis function to approximate $X(t)$,

$$X(t) = s(t)^T\boldsymbol{c}, \boldsymbol{c} \sim N(\boldsymbol{\mu_c}, \Gamma)$$

then we can give the Functional generalized linear model (FGLM),

$$\begin{aligned} p(y_i;\theta_i,\phi) &= \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right\}, \\ g(\mu_i) &= \beta_o + \int \omega_1(t)X(t)dt \\ &= \beta_o + \int \omega_1(t)s(t)^T\boldsymbol{c_i}dt \\ &= \beta_0 + \beta_1^T\boldsymbol{c}_i. \end{aligned}$$

For functional classification, we usually choose logistic function as link function and Y becomes a Bernoulli variable, then FGLM becomes

$$Y_i = \begin{cases} 1 & with\,probability\,\{1 + \exp(-\beta_0 - \beta_1^T\boldsymbol{c}_i)\}^{-1}, \\ 0 & with\,probability\,\{1 + \exp(\beta_0 + \beta_1^T\boldsymbol{c}_i)\}^{-1}, \end{cases}$$

In general, if $E(Y|\boldsymbol{X}) = P(Y = 1|\boldsymbol{X}) > 0.5$, then a new subject is predicted as 1 and 0 otherwise. For parameter estimation, we first can get the observed data likelihood,

$$\begin{aligned} l(\boldsymbol{\mu_c}, \Gamma, \beta_0, \beta_1, \phi, \sigma_\epsilon^2) = &\sum_{i=1}^{N}\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right\} \\ &-\textstyle\sum_{i=1}^{N}\{\frac{n_i}{2}\log(\sigma_\epsilon^2) + \frac{1}{2\sigma_\epsilon^2}(\boldsymbol{x}_i - S_i\boldsymbol{c}_i)^T(\boldsymbol{x}_i - S_i\boldsymbol{c}_i)\} \\ &-\textstyle\sum_{i=1}^{N}\left\{\frac{1}{2}\log|\Gamma| + \frac{1}{2}(\boldsymbol{c}_i - \boldsymbol{\mu_c})^T\Gamma^{-1}(\boldsymbol{c}_i - \boldsymbol{\mu_c})\right\}. \end{aligned} \quad (3)$$

but for this model we can not optimize directly because $\boldsymbol{c}_i$s are unobserved, so we use the EM algorithm which iterates between a maximization (M-step) and an expectation (E-step) to optimize the observed likelihood.

2.2. Functional Generalized Additive Model

Before introduce Functional Generalized Additive Model (FGAM), we first introduce Generalized Additive Model (GAM). For linear regression with a Bernoulli variable Y and a set of predictor variables $X_1, \dots, X_p$, we have model,

$$Y = g\left(\beta_0 + \textstyle\sum_{j=1}^{p}\beta_jX_j + \epsilon\right), \quad (4)$$

where $g(\cdot)$ is a link function, and GAM replaces the linear function β_jX_j by a non-linear function to get

$$Y = g\left(\beta_0 + \textstyle\sum_{j=1}^{p}f_j(X_j) + \epsilon\right).$$

And FGAM is the extension of FAM to functional predictor, which can be expressed as:

$$Y = g\left(\beta_0 + \textstyle\sum_{j=1}^{p}f_j\left(X_j(t)\right) + \epsilon\right).$$

For the estimation of the FGAM, we can use kernel estimation method to get $f_j(\cdot)$:

$$\widehat{f_j^l}(X_j) = \frac{\sum_{i=1}^{N}(Y_i - \hat{Y}_i^{-j,l})K_j(\frac{d_j(X_j,X_{ij})}{h_j})}{\sum_{i=1}^{N}K_j(\frac{d_j(X_j,X_{ij})}{h_j})}, \quad (5)$$

where $\hat{Y}_i^{-j,l} = \sum_{i=1}^{j-1}\widehat{f_j^l}(X_{i\cdot}) + \sum_{i=j+1}^{p}\hat{f}_i^{(l-1)}(X_{i\cdot})$ is the prediction without variable j, d_j is the distance (induced by the norm) , and K_j and h_j are an asymmetric kernel function and the bandwidth, respectively.

2.3. BP Neural Network

BP neural network is a kind of multi-layer feedforward neural network, whose characteristic are forward signal transmission and error back propagation. In the process of forward propagation, the input

DOI https://doi.org/2024-JICS-22388 | Generated on 2025-04-05 21:50:22

signal is processed from the input layer through the hidden layer to the output layer. Each layer of neuron state affects only the next layer of neuron state. If the output layer cannot get the expected output, it will turn to back propagation, and adjust the network weight and threshold according to the prediction error, so that the predicted output of BP neural network keeps approaching the expected output. Figure 1 shows the topology of BP neural network, where $x_1, x_2, \dots, x_m$ are the input of BP neural network and $y_1, y_2, \dots, y_m$ and $\omega_{ij}, \omega_{jk}$are the prediction and weight of BP neural network.

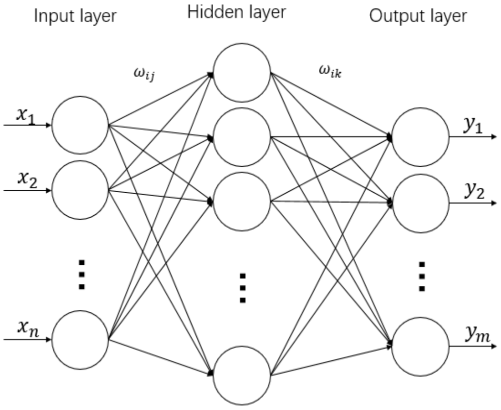

Figure 1. The structure of BP Neural Network

Before prediction, BP neural network should be trained, and the network should be equipped with associative memory and prediction ability through training. The training process of BP neural network includes the following steps:

1. Initial network: Determine the number of nodes in the network input layer N, the number of nodes in the hidden layer L, the number of nodes in the output layer M, initialize the connection weights ω_{ij} and ω_{jk} between the neurons in the input layer, hidden layer and output layer, initialize the threshold a of the hidden layer and the threshold b of the output layer, and give the learning rate and the neuron activation function.
2. Calculate hidden layer output. According to the input variable X, the connection weight ω_{ij} of input layer and the and the threshold a of hidden layer, calculate the output of hidden layer hidden layer H.
$$H_i = f\left(\sum_{i=1}^{n} w_{ij} x_i - a_j\right), j = 1,2, \dots, L.$$
3. Calculate output layer. According to the hidden layer output H, weight ω_{jk} and threshold b, calculate the prediction of BP neural network O.
$$O_k = \sum_{j=1}^{L} H_j \omega_{jk} - b_k, k = 1, \dots, M.$$
4. Calculate error. According to the prediction of network O and expected output Y, calculate the error of network.
$$e_k = Y_k - O_k, k = 1,2, \dots, M.$$
5. Update weight. According to the error, update the weight of network ω_{ij}, ω_{jk}.
6. Update threshold. According to the error, update the threshold a and b.
7. Repeat 2-6 until convergence.

2.4. Support Vector Machine

When given training set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_m, y_m)\}, y_i \in \{-1, +1\}$, what we want to do is to find a hyperplane $\boldsymbol{\omega}^T \boldsymbol{x} + b = 0$ to divide different classifications of samples and the distance from any point $\boldsymbol{x}$ in our sample space to the hyperplane can be expressed as

$$r = \frac{|\boldsymbol{\omega}^T \boldsymbol{x} + b|}{||\boldsymbol{\omega}||}. \tag{6}$$

DOI https://doi.org/2024-JICS-22388 | Generated on 2025-04-05 21:50:22

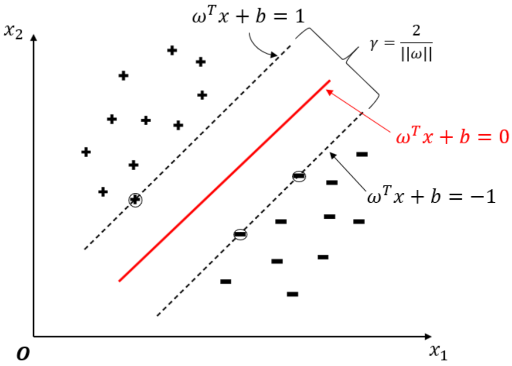

Figure 2. The structure of SVM

Suppose that hyperplane can classify the training set correctly, the we have:

$$\begin{cases} \boldsymbol{\omega}^T \boldsymbol{x}_i + b \geq +1, & y_i = +1; \\ \boldsymbol{\omega}^T \boldsymbol{x}_i + b \leq -1, & y_i = -1, \end{cases}$$

As Figure 2 shows that, the closest point to the hyperplane makes the equal sign established which are called support vector, and the sum of distance from two support vectors which belongs to different classifications to the hyperplane is

$$\gamma = \frac{2}{||\boldsymbol{\omega}||},$$

which is called margin.

To find the hyperplane which has the maximum margin, then we have model:

$$\max_{\boldsymbol{\omega},b} \frac{2}{||\boldsymbol{\omega}||} \tag{7}$$
$$s.t. \, y_i(\boldsymbol{\omega}^T \boldsymbol{x}_i + b) \geq 1, i = 1,2,\dots,m.$$

this model is equivalent to

$$\min_{\boldsymbol{\omega},b} \frac{1}{2} ||\boldsymbol{\omega}||^2 \tag{8}$$
$$s.t. \, y_i(\boldsymbol{\omega}^T \boldsymbol{x}_i + b) \geq 1, i = 1,2,\dots,m.$$

We use Lagrange Multiplier method on equation(8) to get its dual problem, then we have:

$$L(\boldsymbol{\omega}, b, \boldsymbol{\alpha}) = \frac{1}{2} ||\boldsymbol{\omega}||^2 + \sum_{i=1}^m \alpha_i (1 - y_i(\boldsymbol{\omega}^T \boldsymbol{x}_i + b)), \tag{9}$$

where $\boldsymbol{\alpha} = (\alpha_1; \dots; \alpha_m)$, let $\frac{\partial L(\boldsymbol{\omega},b,\boldsymbol{\alpha})}{\partial \boldsymbol{\omega}} = 0$ and $\frac{\partial L(\boldsymbol{\omega},b,\boldsymbol{\alpha})}{\partial b} = 0$, we have:

$$\boldsymbol{\omega} = \sum_{i=1}^m \alpha_i y_i \boldsymbol{x}_i \, , 0 = \sum_{i=1}^m \alpha_i y_i.$$

Finally, we can get the dual problem of equation(8)

$$\max_{\boldsymbol{\alpha}} \sum_{i=1} \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{10}$$
$$s.t. \sum_{i=1}^m \alpha_i y_i = 0, \alpha_i \geq 0, i = 1,2,\dots,m.$$

And we use Sequential Minimal Optimization method to solve equation(10).

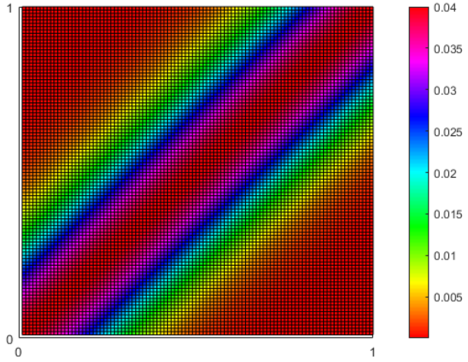

Figure 3. The form of squared exponential kernel function C.

3. Simulation study

DOI https://doi.org/2024-JICS-22388 | Generated on 2025-04-05 21:50:22

In this section, we will design a simulation study to show the performance of the four classification models proposed above. The data which used in this section are generated from the functional mixed-effect model:

$$y_i^{(q)}(t) = \mu^{(q)}(t) + \tau_i(t) + \epsilon_i(t), i = 1, \dots, n, q = 1, \dots, Q$$

where $y_i^{(q)}$ is the observations of the i-th subject which belongs to the q-th classification, $\mu^{(q)}(t)$ is called fixed-effects term which indicates the common characteristics for all the subjects which belongs to the q-th classification, $\tau_i(t)$ is called random-effects term which indicates the own characteristics for the i-th subject and $\epsilon_i(t)$ is called random error term which is used to express the unexplainable situation. We choose the number of classification $Q = 3$ and the three mean functions for fixed-effect term are $\mu^{(1)} = \sin(2\pi t)$, $\mu^{(2)} = 0.8 \times \sin(2\pi t)$, $\mu^{(3)} = 0.7 \times \sin(2\pi t)$, $t \in [0,1]$; For the random-effects, we generate τ_i from Gaussian process $GP(\mathbf{0}, C)$ with the parameter $w = 1, v_0 = 0.04$ in squared exponential kernel function C which has the structure $v_0 \exp\left\{-\frac{w}{2}\left(t_{ij} - t_{ik}\right)^2\right\}$ and the random-error term $\epsilon_i(t)$ is also generated from normal distribution $N(0,0.2^2)$. Figure 1 shows the form of squared exponential kernel function C and Figure 2 shows the three classifications raw data with different colors.

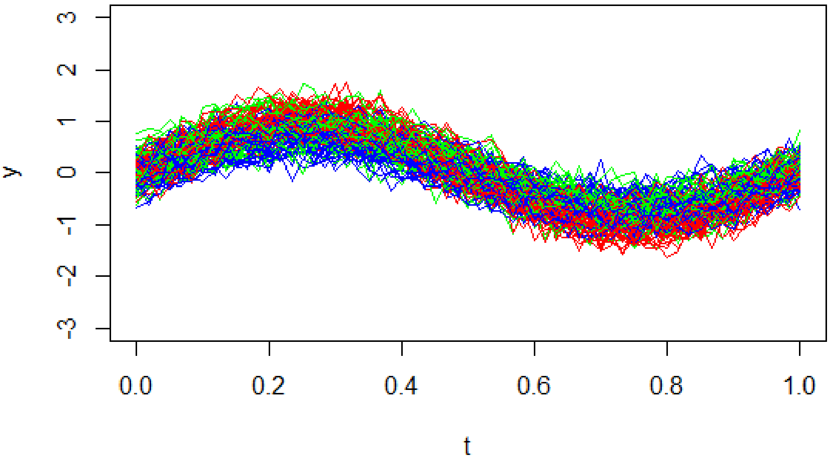

Figure 2. Three classifications raw data with different colors. Red: the first classification with $\mu^{(1)} = \sin(2\pi t)$; Green: the second classification with $\mu^{(2)} = 0.8 \times \sin(2\pi t)$; Blue: the third classification with $\mu^{(3)} = 0.7 \times \sin(2\pi t)$.

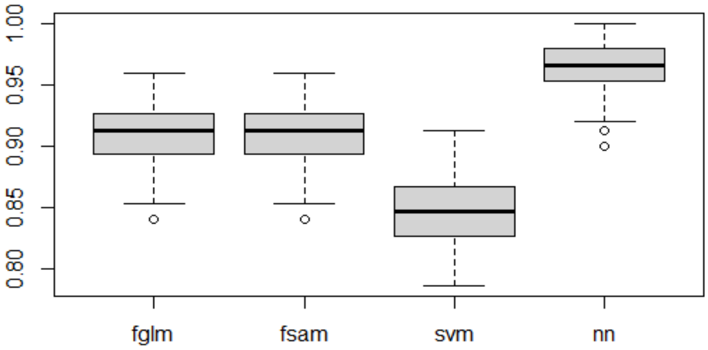

Figure 3. The boxplot of correct rate of four methods in 100 independent simulations.

We repeat 100 independent simulations to get stable results, Table 1 shows the mean and standard deviation of classification correct rate for four methods proposed above in 100 independent simulations. We find that the performance of two functional classification model is almost no difference, the best model is nn whose correct rate is 95.94% and the poorest model is svm whose correct rate is only 84.37%. Figure 3 shows the boxplot of correct rate for four methods in 100 independent simulations which has the same result as what Table 1 shows.

To verify the generalization ability of the model, we generate another 30 new data from functional mixed-effect model for each classification and finally the number of test set is 90. Table 2 shows the prediction correct rate for four methods, we find that the two functional classification methods have the best performance in prediction whose correct rate both are 90.81%, however the prediction correct rate of the best model nn in train set is only 83.50%, which express that the model nn has overfitting phenomenon and the poorest model is still svm.

Table 1. The mean and standard deviation of classification correct rate of four methods proposed above in 100 independent simulations.

Model	Correct rate	Sd
Fglm	90.81%	0.02
Fgam	90.81%	0.02
Svm	84.37%	0.03
Nn	95.94%	0.02

Table 2. The prediction correct rate of four methods.

Model	Correct rate	Sd
Fglm	86.72%	0.04
Fgam	86.73%	0.04
Svm	76.17%	0.05
Nn	83.50%	0.05

Figure 4 shows the boxplot of prediction correct rate for four methods. From the results we find that the two functional model Fglm and Fgam have the best performance on functional data because they can catch the nonlinear structure of functional data which the machine learning model cannot.

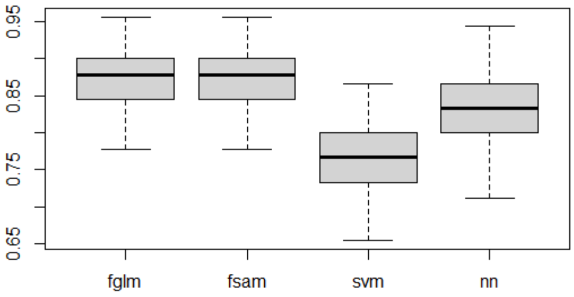

Figure 4. The boxplot of prediction correct rate for four methods.

4. Conclusion

Functional classification is a popular region in functional data analysis. In this paper, we introduce four models based on classification. To compare the performance of the four models, we design simulation study, in the training set the best model which has the highest correct rate is nn, the two functional classification models are Followed by it and the svm is the worst model. However, in the test set, the nn model is not as effective as the two functional models which indicates that the nn model is overfitted, and the latter have better generalization ability.

References

[1] Li, B., Zhang, X. & Smerdon, J. E. Comparison between spatio-temporal random processes and application to climate model data. Environmetrics, 2016, 27(5): 267-279.

[2] Cao, C. Z., Chen, M. Q., Zhu, X. X., et al. Bayesian inference in a heteroscedastic replicated measurement error model using heavy-tailed distributions. Journal of Statistical Computation & Simulation, 2017, 87(15): 1-14.

[3] Chiou, J. M. & Li, P. L. Functional clustering and identifying substructures of longitudinal data. Journal of the Royal Statistical Society: Series B, 2007, 69(4): 679-699.

[4] Zhu, H. X., Brown, P. J., & Morris, J. S. Robust, adaptive functional regression in functional mixed model framework. Journal of the American Statistical Association, 2011, 106(495): 1167-1179.

[5] Hyun J W , Li Y , Huang C , et al. STGP: Spatio-temporal Gaussian process models for longitudinal neuroimaging data. NeuroImage, 2016, 134:550-562.

[6] Chiou,J.M. Dynamical functional prediction and classification, with application to traffic flow prediction. The Annals of Applied Statistics, 2012, 6(4):1588-1614.

[7] Ferraty F , Vieu P . Curves discrimination: a nonparametric functional approach. Computational Statistics & Data Analysis, 2003, 44(1-2):161-173.

[8] Li, B. and Yu, Q. Classification of functional data: A segmentation approach. Computational Statistics & Data Analysis, 2008, 4790-4800.

[9] Goldsmith J , Bobb J , Crainiceanu C M , et al. Penalized Functional Regression. Journal of Computational and Graphical Statistics, 2011, 20(4):830-851.

[10] Shouxiang, Wang, Na, et al. Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method. Renewable Energy, 2016.

[11] Gayathri N , K. Mahesh. Improved Fuzzy-Based SVM Classification System Using Feature Extraction for Video Indexing and Retrieval. International Journal of Fuzzy Systems, 2020:1-14.