Software for classification of Generalized Functional Data

Anthony Weishampel[1], **Ana-Maria Staicu**[1], **and William Rand**[2]

[1]**Department of Statistics, North Carolina State University**
[2]**Poole College of Management, North Carolina State University**

ABSTRACT

In this paper, we describe the developed software package to model and classify binary-valued functional data. The developed software is able to model binary data through the approaches presented in the previous parts. It also provides additional model-based discrimination methods, which can be implemented for these data structures. Additional estimation methods are derived for extreme cases. The developed functions and their arguments are presented.

Keywords: Multilevel, Functional Data, Binary Series, Social Media

1 INTRODUCTION

Binary-valued functional data are sequences of 0/1 values observed at various points within a compact closed interval. These data are common in social media (Weishampel et al., 2021a), accelerometers (Goldsmith et al., 2015), ecological occupational models (Johnson et al., 2013), spatial light indicators (Serban et al., 2013), and drug use records (Ye et al., 2015). Several parametric approaches have been explored to model these data, such as generalized estimating equations (GEEs) (Liang & Zeger, 1986; Lipsitz et al., 1991; Rao Chaganty & Joe, 2004), generalized linear mixed models (GLMMs) (Heagerty, 1999; Breslow & Clayton, 1993) and other longitudinal data analysis approaches (Diggle et al., 2002).

Hall et al. (2008) introduced a new nonparametric functional data analysis approach for modeling the binary data observed at a few time points. Hall et al. (2008), expanded upon the real-valued functional principal analysis of Yao et al. (2005) to be applied to binary-valued functional data and assumed that the binary data are realizations of a latent continuous process. This allowed the binary-valued functional data to be modeled by a Karhunen-Loève (KL) representation of the latent curves. These functional data analysis based methods for binary-valued functional data have been further explored for instances when the binary data are observed at many time points (McLean et al., 2014; Wood et al., 2016), have a multi-level structure (Goldsmith et al., 2015), or have a spatial correlation structure Serban et al. (2013). Despite these methods to model the data; few methods have been developed for classifying binary-valued functional data (Weishampel et al., 2021a).

Many packages have been developed to handle real-valued functional data; however, few exist for binary-valued functional data. The fundamental **FDA** package is a standard *R* package for modeling functional data through parametric approaches (Ramsay et al., 2012) . Ferraty & Vieu (2006) is outlines many nonparametric approaches for real-valued functional data and the accompanying *R* code for these methods are available at http://www.lsp.ups-tlse.fr/staph. Similarly, many other packages in R have been developed for various other different scenarios of real-valued functional data. For example **fpca** implements a restricted maximum likelihood approach to estimate the functional principal component analysis (FPCA) (Peng & Paul, 2011), **ftsa** accounts for functional time series analyses, and **fdapace** estimates FPCA for sparse data through the Principal Analysis by Conditional Estimation (PACE) algorithm (Gajardo et al., 2021). Software have been developed for classifying functional data, as **fda.usc** package provides parametric functional approaches (Bande et al., 2020) and the accompanying code of Ferraty & Vieu (2006) presents nonparametric classification methods. While all of these methods have been developed for real-valued functional data, none of these packages are able to account for binary-valued functional data.

A few R packages have been developed for generalized functional data. For example **mgcv** implements generalized additive mixed models (Wood, 2021) and **refund** presents regression methods for functional data (Goldsmith et al., 2020). Both of these packages can be used to model binary-valued functional data. Similarly the accompanying code

provided by (Goldsmith et al., 2015) can be used to model the binary data for multi-level instances. Goldsmith et al. (2015) presents a Bayesian approach, coded in STAN, to model data where there are multiple independent series observed for each individual.

However, these packages do not include methods to model the data through generalized functional principal component analyses (gFPCA), as presented in Hall et al. (2008) and Serban et al. (2013). Additionally, these methods run into computational complications for large data sets, especially when the data are as large as some social media datasets. For scenarios when there are multiple binary series observed per individual, the multi-level models fail to account for scenarios when there can be series with no positive observations. Most importantly, none of these packages provide classification methods for the binary data. There are many instances, when the researchers would like to discriminate and identify group structures for the binary data. For example, in social media, malicious and different types of accounts can be detected based on their binary data indicating time periods when account's are posting and not posting (Overgoor et al., 2019; Weishampel et al., 2021a,b). Thus, there is a clear gap in the available software for modeling and classifying binary-valued functional data.

In this paper, we present and describe the R package **gFPCAclassif**, which models and trains classifiers for binary-valued functional data. The package presents model-based classification methods for the binary-valued functional data. The package accounts for multiple different scenarios of the data. All of these models are built under the assumption that the binary data are binary-valued observations of smooth latent functions. This package is currently available on GitHub at https://github.com/acweisha/gFPCAClassif. The package differentiates itself from other software as it is the first method to model and discriminate the binary-valued functional data through gFPCA methods. As observed in the analyses in Weishampel et al. (2021a), these methods are able to discriminate the binary-valued functional data with higher accuracy than many alternatives, while providing interpretable models.

Throughout this paper and in the package, we assume that the binary-values are observed at points, which are equispaced and regular throughout the closed compact interval. This assumption is valid for most instances when the data are recorded near real time, e.g. social media, accelerators, and health monitors.

The paper is organized as follows. Section 2 overviews the three models of the binary-valued functional data, which are addressed in the package. Section 3 introduces new estimation procedures introduced for the package. Section 4 presents the main functions of the package and explains the arguments of the functions. Section 5 provides a practical application of the package and guides the reader through an example of how to implement the package. Section 6 concludes with final remarks.

2 GENERALIZED FUNCTIONAL CLASSIFICATION MODELS

In this section we present the modeling and discrimination methodology for the functional data analysis based models. After a brief introduction of the discrimination methodology,

we describe the most important aspects, features, and utilities of models. Let Y be the variable indicating the group, where there are L groups $Y \in \{0, ..., L-1\}$. Let $\mathbf{X}$ be the generic comprehensive expression of a subject's binary functional data and $\mathbf{C}$ be its covariate information. Additionally, consider a new subject's binary data $\boldsymbol{x}$ and covariates $\mathbf{c}$. Using this notation, a new individual is classified to the group or class that maximizes the probability, $Y_x = \underset{l \in \{0,1,...,L-1\}}{\arg\max} \; P(Y = l | \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{C} = \boldsymbol{c})$. All of the classification methods of the package are based upon a nonparametric Bayesian classification rule. The discrimination rule is further calculated as

$$P(Y = l | \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{C} = \boldsymbol{c}) = \frac{P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{C} = \boldsymbol{c} | Y = l)\pi_l}{\sum_{l'=0}^{L-1} P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{C} = \boldsymbol{c} | Y = l')\pi_{l'}} \tag{1}$$

and $\pi_l = P(Y = l)$ represents the a priori probability of the group "l". As presented in the introduction, this package considers three different scenarios of binary-valued functional data. In each one of the subsequent scenarios the classifiers are developed based upon this Bayes classifier.

2.1 gsFPCA: Generalized Single-level FPCA

The first scenario presents a discrimination model, when there is one binary-valued function observed for each individual. Let i index the n users, $i = 1, ..., n$, and let the data for the i^{th} user be $\{(X_{ip}, t_{ip})_{p=1}^{m}, \mathbf{C}_i, Y_i\}$ consisting of a binary series $X_{ip} \in \{0, 1\}$, corresponding to a fine time grid $t_{ip} \in \mathcal{T}$, for $p = 1, \ldots, m_i$, for large m_i, class membership $Y_i \in \{0, ..., L-1\}$, and $\mathbf{C}_i$ is Q-dimensional vector of covariates. We assume that $\mathcal{T}$ is a compact interval and that the covariates are independent from the binary functional data. We assume that for each user i, the observed binary series $(X_{ip}, t_{ip})_{p=1}^{m_i}$ is a realization of the binary-valued function $X_i : \mathcal{T} \to \{0, 1\}$. The binary data are observed at m_i time points $t_{i1}, \ldots, t_{im_i}$; with $X_{ip} = X_i(t_{ip})$ for all $p = 1, ..., m_i$. Let $Z_i(\cdot)$ be a latent continuous process for account i. Having defined this notation, we posit the generalized single-level model for the observed binary data $X_{ip} = X_i(t_{ip})$ given the group membership $Y_i = l$:

$$\begin{aligned} &X_{ip} | Z_i(\cdot), t_{ip} \overset{iid}{\sim} Bernoulli\{p_i(t_{ip})\}, \qquad p = 1, \ldots, m_i \\ &g^{-1}\{p_i(t)\} = Z_i(t), \;\; t \in [0, 1], \qquad i = 1, \ldots, n \\ &Z_i(\cdot) | Y_i = l \sim GP(\mu_l, \Sigma_l), \;\; l = 0, 1, \ldots, L-1 \end{aligned} \tag{2}$$

where $GP(\mu_l, \Sigma_l)$ denotes a Gaussian process with group dependent mean function $\mu_l(\cdot)$ and group dependent covariance function $\Sigma_l(\cdot, \cdot)$. To account for the binary data, $g(\cdot)$ is the logistic link function with $g^{-1}(\cdot)$ denoting its inverse.

Similar to Hall et al. (2008), the latent process $Z_i(\cdot)$ is approximated through a the finite truncation of the Karhunen-Loève (KL) representation, by $Z_i(t) \approx \mu(t) + \sum_{k=1}^{K} \xi_{ik}\phi_k(t)$, where $\mu(t) = E\{Z_i(t)\}$, $\{\phi_k(\cdot)\}_{k \geq 1}$ are the eigenfunctions of the covariance function $\Sigma(t, t') = Cov\{Z_i(t), Z_i(t')\}$, and $\xi_{ik} = \int_{\mathcal{T}} \{Z_i(t) - \mu(t)\}\phi_k(t)dt$

are functional principal components scores. The functional principal components scores have zero mean and variance equal to λ_k, where λ_k is the k^{th} largest eigenvalue in the decomposition of $\Sigma(t, t')$.

As explored in previous methods of classifying functional data, the probability of the $P\{Z(\cdot) = z(\cdot)|Y = l\}$ can be approximated using the leading principal components scores in the finite approximation (Delaigle & Hall, 2012; Dai et al., 2017; Huang & Ruppert, 2019). This idea of considering the K leading principal component for classification is extended to the case of binary-valued functional data. Using the truncated representation of the latent trajectory and the independence assumption of between the binary-valued functional data and the covariates, the Bayes classifier results in the discrimination rule of

$$Y_x = \arg\max_l P(Y = l|\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{C} = \boldsymbol{c}) \quad \approx \quad \arg\max_l \{f_l(\boldsymbol{\xi}) f_{l,c}(\mathbf{c})\pi_l\}, \tag{3}$$

where $f_l(\cdot)$ denotes the group dependent joint distribution of the K leading principal component scores, $\boldsymbol{\xi}$, and $f_{l,c}(\cdot)$ is the group dependent joint distribution of the covariates for group l. To fit this classifier in equation 3, we need to obtain the Karhunen-Loève approximation of the latent curves for each binary-valued functional realization, the prior probability, and the group dependent densities of the principal component scores and additional covariates.

Recall, the KL approximation requires estimates for the mean function, eigenfunctions and the principal component scores. Estimates for obtaining the mean function and eigenfunctions are further described in Section 3. After estimating the mean function and eigenfunctions the principal component scores are recovered by fitting a generalized mixed linear model around these estimates,

$$\begin{aligned} &X_{ip}|\xi_{i1}, \ldots, \xi_{iK}, t_{ip} \sim Bernoulli(p(t_{ip})),\ p = 1, \ldots, m_i \\ &g^{-1}\{p(t)\} = \widehat{\mu}(t) + \widehat{\phi}_1(t)\xi_{i1} + \ldots + \widehat{\phi}_K(t)\xi_{iK} \\ &\xi_{ik} \sim N(0, \widehat{\lambda}_k), k = 1, \ldots, K; \text{ and } \xi_{i1}, \ldots, \xi_{iK} \text{ are independent.} \end{aligned} \tag{4}$$

The principal component scores are assumed to be independent and the univariate estimates of the distributions are estimated using a kernel density estimator. To estimate the univariate marginal densities of the scores, we consider kernel-based density estimators (Silverman, 1986). Let $\mathbb{K}(\cdot)$ be a univariate kernel function such as Gaussian or Epanechnikov; the kernel density estimate of the conditional expectation of the kth basis coefficients in class l is

$$\widehat{f}_{lk}(u) = \frac{1}{n_l h_{lk}} \sum_{i:Y_i=l} \mathbb{K}\left(\frac{u - \widehat{\xi}_{\boldsymbol{X},ik}}{h_{lk}}\right), u \in \mathbb{R}, \tag{5}$$

where $h_{lk} = h\widehat{\sigma}_{lk}$ is bandwidth corresponding to the estimating the distribution of the conditional expectation scores of the kth eigenfunction and class l,$\{\widehat{\xi}_{\boldsymbol{X},ik} : Y_i = l\}$, and $\widehat{\sigma}_{lk}$ is the corresponding estimated standard deviation. The bandwidth multiplier h is

selected through a grid search of possible values minimizing the misclassification rate in 5-fold cross validation (CV) of the training set. The reader can consult Weishampel et al. (2021a) for more details. The package currently models all of the components through Gaussian kernels.

Similar nonparametric methods are used when estimating the joint density of the additional covariates $f_{l,c}(\cdot)$. The methods of the package are able to account for both continuous numerical and categorical variables. The distributions of numerical covariates are estimated using the same nonparametric kernel based methods as presented Equation 5. Specifically, for each of these continuous covariates the density conditioned by group is approximated as

$$\widehat{f}_{lc_q}(u) = \frac{1}{n_l h_{lq}} \sum_{i:Y_i=l} \mathbb{K}\left(\frac{u - c_{iq}}{h_{lq}}\right), u \in \mathbb{R},, \tag{6}$$

where $h_{lq} = h\widehat{\sigma}_{lq}$ is its corresponding bandwidth for the kernel estimator, $\widehat{\sigma}_{lq}$ is estimated standard deviation of q^{th} covariates for individuals in group l, and the scaling parameter h is defined as before. For each categorical covariates, the probability mass functions are estimated by the observed density within each group, i.e. $\widehat{f}_{lc_q}(u) = n_l^{-1} \sum_{i:Y_i=l} \mathbb{1}_{(u=c_{iq})}$ where $n_l = \sum_{i=1}^{n} \mathbb{1}_{(Y_i=l)}$. This is a common approach for categorical variables in Bayes classifiers (Hsu et al., 2008). Using the density estimates, the updated Bayes classifier is

$$\widehat{y} = \arg\max_l \widehat{\pi}_l \widehat{f}_l(\widehat{\boldsymbol{\xi}}) \prod_{q=1}^{Q} \widehat{f}_{l,c_q}(c_q) \tag{7}$$

where $c_1, ..., c_q$ are the observed covariates for a new individual, $\widehat{f}_l(\widehat{\boldsymbol{\xi}}) = \prod_{k=1}^{K} \widehat{f}_{lk}(\widehat{\xi_k})$, and $\widehat{\pi}_l = n_l/n$.

2.2 gmFPCA: Generalized Multi-level FPCA

The second model should be utilized when there are multiple observed binary series for each individual and each binary series observed in $\mathcal{T}$. Let i index the n users, $i = 1, ..., n$, and let the data for the i^{th} user be $\{(X_{i1p}, t_{i1p})_{p=1}^{m}, \ldots, (X_{iJ_ip}, t_{iJ_ip})_{p=1}^{m}, \mathbf{C}_i\ Y_i\}$, where $\{(X_{ijp}, t_{ijp})_{p=1}^{m}\}$ is the jth observed binary-valued functional data, $X_{ijp} \in \{0, 1\}$, $t_{ijp} \in \mathcal{T}$, for $p = 1, \ldots, m$. Similar to the gsFPCA model, we assume independence between the covariates and the binary data.

We assume that for every i and j the binary-valued functional data are realizations of a latent continuous curve, $Z_{ij}(\cdot)$. In this regard we, propose the following model for the observed binary data:

$$\begin{aligned} &X_{ijp}|Z_{ij}(\cdot), t_p, \overset{iid}{\sim} Bernoulli\{p_{ij}(t_p)\}, \qquad p = 1, \ldots, m \\ &g^{-1}\{p_{ij}(t)\} = Z_{ij}(t), \ \ t \in [0, 1] \end{aligned} \tag{8}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, J_i$, where $Z_{ij}(\cdot)$ is the latent curves that describes the binary data of individual i on the j^{th} curve, and $g(\cdot)$ is the logistic link function.

Additionally, we assume that for every i and j that the latent curves contains large values and the number of observed time points m is large enough such that the $P(\sum_{p=1}^{m} X_{ijp} = 0)$ is negligible. This assumption ensures that positive values are observed for each binary series.

Because of the repeated nature of binary functional data, a multi-level approach similar to the ones presented in Serban et al. (2013) and Goldsmith et al. (2015) is used to model latent signal. Specifically, the latent process $Z_{ij}(\cdot)$ is modeled as

$$\begin{aligned} &Z_{ij}(t) = V_i(t) + W_{ij}(t), \\ &V_i|Y_i = l \overset{indep}{\sim} \text{GP}(\mu_l, \Sigma_l) \\ &W_{ij} \overset{iid}{\sim} \text{GP}(0, \Sigma_W). \end{aligned} \tag{9}$$

where $V_i(\cdot)$ is the individual process, and $W_{ij}(\cdot)$ is the within individual curve specific process. Both of these functional effects are defined on $t \in \mathcal{T}$ and are assumed to be independent from each other. Similar to the latent functions approximations in gsFPCA, both latent signals are approximated through a finite truncation of the KL representation. Specifically, the individual effect and the specific within individual curve signal are approximated as $V_i(t) \approx \mu(t) + \sum_{k}^{K_V} \xi_{ik}\phi_{V,k}(t)$ and $W_{ij}(t) \approx \sum_{k}^{K_W} \zeta_{ijk}\phi_{W,k}(t)$ respectively, where $\mu(\cdot)$ the mean function, $\{\phi_{V,k}(\cdot), \lambda_{V,k}\}_{k\geq 1}$, $\{\phi_{W,k}(\cdot), \lambda_{W,k}\}_{k\geq 1}$ are the eigenfunctions and eigenvalues of the covariance functions $\Sigma_V(t,t') = Cov\{V(t), V(t')\}$ and $\Sigma_W(t,t') = Cov\{W(t), W(t')\}$. The functional principal components scores, ξ_{ik} and ζ_{ijk}, are zero-mean random variable mutually uncorrelated, and variances equal to $\lambda_{V,k}$ and $\lambda_{W,k}$, respectively. Resulting in the latent curves being approximated as $Z_{ij}(t) \approx \mu(t) + \sum_{k=1}^{K_V} \xi_{ik}\phi_{V,k}(t) + \sum_{k=1}^{K_W} \zeta_{ijk}\phi_{W,k}(t)$.

The vector of individual level components $\boldsymbol{\xi}_i = (\xi_{i1}, ..., \xi_{iK_V})^T \in \mathbb{R}^{K_V}$ extracts the main features explaining the user's deviation away from the unconditional mean. By once again summarizing, the individual's binary curves into this vector of principal component coefficients we incorporate the scores into the classifier. The Bayes classifier results in the discrimination rule of

$$Y_x = \arg\max_l P(Y = l|\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{C} = \boldsymbol{c}) \quad \approx \quad \arg\max_l \{f_l(\boldsymbol{\xi}) f_{l,c}(\mathbf{c})\pi_l\},$$

where $f_l(\cdot)$ denotes the group dependent joint distribution of $\boldsymbol{\xi}$, the K_V leading principal component scores of the individual effect $V_i(\cdot)$, and $f_{l,c}(\cdot)$ is the group dependent joint distribution of the covariates for group $Y = l$.

Estimating the mean function and eigen-components require a more in depth description, thus the methodology used to obtain them will be presented in section 3. However, once we have obtained these estimates we can predict the individual level coefficients. Using the estimates of the mean function and eigen-components, the account-specific coefficients are predicted by $\widehat{\xi}_{ik} = E\left[\xi_{ik} \,|\{(X_{i1p}, t_{i1p})_{p=1}^{m}\}_{j=1}^{J_i}\right]$. Specifically, the predictions are estimated using a GLMM, where the observed data are modeled by the

estimated mean functions and eigen-components,

$$
\begin{aligned}
&X_{ijp}|\boldsymbol{\xi}_i, \boldsymbol{\zeta}_{ij}, t_p \overset{indep}{\sim} Bernoulli\left\{p_{ij}(t_p)\right\} p = 1, \ldots, m_i; j = 1, \ldots, J \quad (10)\\
&g^{-1}\left\{p_{ij}(t)\right\} = \widehat{\mu}(t) + \sum_{k=1}^{K_V} \xi_{ik}\widehat{\phi}_{V,k}(t) + \sum_{l=1}^{K_W} \zeta_{ijl}\widehat{\phi}_{W,l}(t)\\
&\xi_{ik} \sim N(0, \widehat{\lambda}_{V,k}), k = 1, \ldots, K_V, \zeta_{ijl} \sim N(0, \widehat{\lambda}_{W,l}), l = 1, \ldots, K_W;
\end{aligned}
$$

and we assume that $\xi_{i1}, \ldots, \xi_{iK_V}, \zeta_{ij1}, \ldots, \zeta_{ijK_W}$ are independent. The estimates of the individual-specific coefficients, $\widehat{\xi}$ are used estimate the group-dependent densities and train the classifier. Estimates of the densities for the individual-specific coefficients and additional covariates are obtained using the same nonparametric estimators defined for the *gsFPCA* model in Section 2.1 (Equations 5-7). The density for the individual-specific coefficients need to be updated to reflect the number of components, $\widehat{f}_l(\widehat{\boldsymbol{\xi}}) = \prod_{k=1}^{K_V} \widehat{f}_{lk}(\widehat{\xi}_k)$.

2.3 gmFPCA+gAR: Generalized Multi-level FPCA + Generalized Autoregression

The third model is an extension from the previous model, where there are realizations of the binary-valued functional data when an individual is inactive throughout the compact interval for $\mathcal{T}$. More precisely, there can be values of j such that the probability, $P\{\sum_{p=1}^{m} X_{ij}(t_{ijm}) > 0\} = 0$. Thus, the data have the same notation as in Section 2.2 and we observe data $\{(X_{i1p}, t_{i1p})_{p=1}^{m}, \ldots, (X_{iJ_ip}, t_{iJ_ip})_{p=1}^{m}, \mathbf{C}_i\ Y_i\}$, for each individual. This model introduces the latent indicator variable, S_{ij} , where if $S_{ij} = 0$ then the individual is not active and $P(X_{ijp} = 1|t_p) = 0$ for all t. Using these indicator variables, we posit the following model for X_{ijp}'s:

$$
\begin{aligned}
&X_{ijp}|Z_{ij}(\cdot), t_p, S_{ij} \overset{iid}{\sim} Bernoulli\{S_{ij}p_{ij}(t_p)\}, \qquad p = 1, \ldots, m\\
&g^{-1}\left\{p_{ij}(t)\right\} = Z_{ij}(t), \ \ t \in [0, 1]
\end{aligned}
$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, J_i$, where the latent process $Z_{ij}(\cdot)$ is modeled using a similar multilevel framework to the *gmFPCA*. Like the previous models $g(\cdot)$ is the logistic link function. We assume that the indicator variables, S_{ij}, and the latent curves $Z_{ij}(\cdot)$ are independent from each other. The latent function, $Z_{ij}(\cdot)$, is further modeled using the same multi-level structure as introduced in the previous section, $Z_{ij}(t) = V_i(t) + W_{ij}(t)$. Using KL approximations to represent signals in the multilevel framework, the latent curves are approximated as $Z_{ij}(t) \approx \mu(t) + \sum_{k=1}^{K_V} \xi_{ik}\phi_{V,k}(t) + \sum_{k=1}^{K_W} \zeta_{ijk}\phi_{W,k}(t)$, where the mean function, various eigenfunctions and principal components are all defined in Section 2.2. The indicators of activeness S_{ij} are modeled over j as

$$
\begin{aligned}
&S_{ij}|Y_i = l,\ S_{i(j-1)}, \ldots, S_{i1} \overset{indep}{\sim} Bernoulli(\alpha_{ij}^{l}),\\
&g^{-1}(\alpha_{ij}^{l}) = \beta_{0,l} + \beta_{1,l}S_{i(j-1)} + \ldots + \beta_{q,l}S_{i(j-q_l)}
\end{aligned}
$$

where the indicators of active periods are modeled through a group dependent generalized autoregressive (gAR) model, with a group dependent lag of q_l and unknown parameters $\boldsymbol{\beta}_l = (\beta_{0,l}, \ldots, \beta_{q,l})$. In this model, we assume that both the periods when the individuals are active and binary observations at times within the active periods both are helpful for discrimination. Using the generalized multi-level functional principal component analysis and generalized autoregressive model framework, we estimate the class of the user,

$$Y_x \quad = \quad \arg\max_l \left\{ P(S_1 = \widetilde{s}_1, \ldots, S_J = \widetilde{s}_J | Y_{\boldsymbol{x}} = l) f_l(\boldsymbol{\xi}) f_{l,c}(\mathbf{c}) \pi_l \right\},$$

where $\widetilde{s}_j = 1(\sum_{p=1}^{m} x_{jp} > 0)$, the group dependent joint densities are defined as before, and the active period probability is defined as

$$P(S_1 = \widetilde{s}_1, \ldots, S_J = \widetilde{s}_J | Y_{\boldsymbol{x}} = l) = P(S_1 = \widetilde{s}_1, \ldots, S_q = \widetilde{s}_q | Y_{\boldsymbol{x}} = l) \times \prod_{j=q+1}^{J} P(S_j = \widetilde{s}_j | Y_{\boldsymbol{x}} = l, S_{j'} = \widetilde{s}_{j'}; j' = j-p, ..., j-1).$$

Predictions of the individual-specific coefficients are obtained using an updated GLMM. The GLMM is very similar to the one present in Equation 11. However, the new GLMM introduces a condition rule, where that it only considers data from realizations when $\widetilde{S}_{ij} = 1$, where $\widetilde{S}_{ij} = 1(\sum_{p=1}^{m} X_{ijp} > 0)$. By omitting the binary-functional data from inactive realizations, individual-specific coefficients are predicted by $\widetilde{\xi}_{ik} = E[\xi_{ik} | \boldsymbol{X}_{ij}$ for all j such that $\widetilde{S}_{ij} = 1]$. The model is further described in Weishampel et al. (2021b). Similar to the *gmFPCA* model, the densities estimates for the individual-specific coefficients and additional covariates are estimated using the kernel approximations in Section 2.1 (Equations 5-7).

By including the autoregressive structure on the latent states, the model introduces another way covariates can be included in this framework. By assuming that the additional covariates affect the latent state, the covariates are included as linear effects in the gAR Weishampel et al. (2021a). Because we assume that the latent state is independent of the latent process these covariates do not affect the estimation of the account-specific coefficients.

Fitting these three models and predicting groups for new individuals are the main focus of the package. However, there are still many different features of the models which are need to be further explained when estimating the mean functions, eigenfunctions, and covariates.

3 ESTIMATION METHODOLOGY

Much of the estimation procedures for the classifiers and the various features needed to fit the classifiers were discussed in the previous sections. However, the previous section omits the estimation methodology used to obtain estimated for the mean function and eigenfunction bases. The procedures to estimate these functions are presented here for

both the single-level and multi-level models. In the multi-level scenario, the estimation methodology differs based on the prevalence of events. More specifically, in the multi-level models the estimates for the mean function and eigenfunction bases are determined by approximating the various covariance functions. The approximation which should utilized differs based on whether the events are rare or not. For the binary data events are considered rare if $g^{-1}\{\mu(\cdot)\}$ is close to 0 or 1.

3.1 Single-level Estimation

As outlined in Weishampel et al. (2021a), estimating the mean function and the classifier consists of two main steps. The two step procedure begins by estimating individuals' smooth latent curves using a known set of basis functions. The second set estimates the mean function and eigenfunctions based on the estimated latent trajectories, by pooling the latent trajectories of across all of the users. Both of these steps includes tuning parameter and features, which the user needs to consider when developing the classifier.

In the first step the latent smooth trajectory is estimated using a predetermined set of basis functions (Scheipl et al., 2016). Let $\{B_k(t) : k = 1, 2, \ldots, K_B, t \in \mathcal{T}\}$ be the preset basis system consisting of K_B functions defined on $\mathcal{T}$. The basis functions approximate the latent trajectory as $Z_i(t) \approx \sum_{k=1}^{K_B} \beta_{ik} B_k(t)$. The unknown basis coefficients $\boldsymbol{\beta}_i = \{\beta_{i1}, ..., \beta_{iK_B}\}^T$ are estimated using a penalized likelihood function, where the penalization parameter is used to ensure smoothness in the estimated trajectory. The type of basis functions and the number of basis functions used are features the researcher needs to consider when estimating these latent trajectories. The type of basis functions will change based on the data. The researcher will want to select the number and type of basis functions which are best able to capture the complexities in the binary-valued functional data; see also Scheipl et al. (2016) and Weishampel et al. (2021a). Using the estimated basis coefficients, the latent trajectories for each user are defined as $\widehat{Z}_i(t) = \sum_{k=1}^{K_B} \widehat{\beta}_{ik} B_k(t)$.

In the second step, the mean function and eigenfunctions are estimated by applying a standard functional principal component analysis for smooth curves on the recovered latent trajectories (Ramsay & Silverman, 2005; Xiao et al., 2013). Specifically, the estimated mean and covariate functions are recovered by applying smoothers across the estimators $\widetilde{\mu}(t) = \sum_{i=1}^{n} \widehat{Z}_i(t)/n$ and $\widetilde{\Sigma}(t, t') = \sum_{i=1}^{n} \widehat{Z}_i(t)\widehat{Z}_i(t')/(n-1)$, respectively. Specially, we apply a one-dimensional smoother to obtain the smoothed estimated mean $\widehat{\mu}(t)$. A bivariate smoother is applied to estimate the covariance function, where we ensure that the resulting estimated smoothed covariate function is symmetric and positive semidefinite. We estimate the eigen-components $\{\widehat{\phi}_k(t), \widehat{\lambda}_k\}_k$ from the spectral analysis of the estimated smooth covariance function. The number of eigenfunctions in the finite truncation, K is determined based on the proportion of explained variance (PVE) which is calculated from the estimated eigenvalues (Di et al., 2009; Hall et al., 2001).

3.2 Multi-level Estimation

As noted in the beginning of this section, the estimation methodology for obtaining the mean function and sets of eigenfunctions in the multi-level models differ based on the

frequency of outcomes. The first estimation methods are developed to be utilized for the scenario when the $g^{-1}\{\mu(\cdot)\}$ is not close to 0 or 1. Since these probabilities are not close to the extreme values, the realizations of the observed binary-valued functional data contain observations with both values 0 and 1. Therefore both of possible outcomes are considered non-rare events. In this instance, when the outcomes are considered non-rare event the linear approximation estimation methodology should be utilized in when fitting the models. The linear approximation is derived by applying a Taylor expansion around the link function $g(\cdot)$.

Conversely, when the data are less balanced and $g^{-1}\{\mu(\cdot)\}$ is close to 0 or 1, one of the events occurs much less frequently. Thus, we refer to this scenario as having rare events. The linear approximation is not accurate for this scenario. For these instances, Serban et al. (2013) presents methodology to estimate the mean functions and various eigenfunctions of the multi-level model by using an exponential approximation. The exponential approximation applies a Taylor approximation to the link function $g(\cdot)$, and assumes that $\exp(x)$ is small enough, such that $\exp(x)/(1+\exp(x)) \approx \exp(x)(1-\exp(x))$.

The rest of this section presents the derivations and estimation procedures to obtain the mean function and eigen-components for these two scenarios of rare and non-rare events. The methods are very similar for the *gmFPCA+gAR* and the *gmFPCA* models. The derivations for the *gmFPCA+gAR* are obtained by conditioning that the realization of the binary-valued functional data is from an active state, $S = 1$. Since the *gmFPCA* model can be considered as the *gmFPCA+gAR* model with $S = 1$ for all realizations, the derivations for the *gmFPCA* from the *gmFPCA+gAR* methods are trivial. Therefore we will only present the methods and derivations for the *gmFPCA+gAR*. The derivations for the *gmFPCA* model are included in the supplemental material.

3.2.1 Non-rare Events: Linear Approximation

When fitting the multilevel models and assuming that the probability of $g^{-1}\{\mu(\cdot)\}$ is not extreme, we estimate the mean and covariance functions, by using a linear approximation of the link function (Serban et al., 2013; Weishampel et al., 2021a). Assuming that the variation of $Z_{ij}(\cdot)$ around its mean is small, we approximate $g\{Z_{ij}(t)\}$ via a Taylor expansion

$$\begin{aligned} g\{Z_{ij}(t)\} \quad &\approx \quad g\{\mu(t)\} + g'\{\mu(t)\}\{V_i(t) + W_{ij}(t)\} + \\ &\qquad \frac{1}{2} g''\{\mu(t)\}\{V_i(t) + W_{ij}(t)\}^2, \end{aligned}$$

where $g'(t) = \partial g(t)/\partial t$ and $g''(t) = \partial^2 g(t)/\partial t^2$. Omitting the term with the second derivative of provides a linear approximation for the marginal probability (Serban et al., 2013). By only including the first two terms of the approximation and conditioning only on active realizations, we obtain $E\{X_{ij}(t) = 1 | S_{ij} = 1\} \approx g\{(\mu(t)\}$. As outlined in Weishampel et al. (2021b), by conditioning only on active realizations the methodology closely follows methods presented in Serban et al. (2013). For these approximations we

utilize the latent state estimator $\widetilde{S}_{ij} = \mathbb{1}_{\sum_{p=1}^{m} X_{ijp}>0}$. This estimator is accurate as long as m and $Z_{ij}(\cdot)$ are large (Weishampel et al., 2021b).

Using the linear approximation the mean function is recovered by $\widehat{\mu}(t) = g^{-1}\{\widehat{p}(t)\}$ where $\widehat{p}(t)$ is the estimated the probability function and obtained through applying a univariate smoother onto the data

$$\left\{ \left(t_p, \frac{\sum_{i=1}^{n}\sum_{j=1}^{J_i} \widetilde{S}_{ij} X_{ijp}}{\sum_{i=1}^{n}\sum_{j=1}^{J_i} \widetilde{S}_{ij}} \right) : p = 1, \ldots, m \right\}. \tag{11}$$

This estimate is obtained by implementing a working assumption of independence across i and j.

To estimate the various covariance functions and the eigenfunctions, the linear approximation is applied to the conditional expectation of the product $X_{ijp}X_{ij'p'}$ conditioned on that the data are both from active realizations,

$$\begin{aligned} E(X_{ijp}X_{ij'p'} = 1 | S_{ij}S_{ij'} = 1, t_p, t_{p'}) \quad &\approx \quad g\left\{\mu(t_p)\right\} g\left\{\mu(t_{p'})\right\} + \\ &\quad \left[\Sigma_V(t_p, t_{p'}) + \Sigma_W(t_p, t_{p'})\mathbb{1}_{j=j'}\right] \times \\ &\quad g'\left\{\mu(t_p)\right\} g'\left\{\mu(t_{p'})\right\} \end{aligned}$$

where $\Sigma_V(t,t') = Cov\{V(t), V(t')\}$, $\Sigma_W(t,t') = Cov\{W(t), W(t')\}$, and $\mathbb{1}_{(j=j')}$ is the indicator for $j = j'$. Continuing to use the working assumption of independence independence across i and j, estimates of the two covariance functions, $\Sigma_V(\cdot,\cdot)$ and $\Sigma_W(\cdot,\cdot)$, are obtained in following steps. We use the estimators method of moment estimators $\widetilde{E}_W$ and $\widetilde{E}_V$ of Weishampel et al. (2021b) to estimate the conditional expectations $E\left[\left(X_{ijp} - X_{ij'p}\right)\left(X_{ijp'} - X_{ij'p'}\right) | S_{ij}S_{ij'} = 1, t_p, t_{p'}\right]$ and $E\left[X_{ijp}X_{ij'p'} | S_{ij}S_{ij'} = 1, t_p, t_{p'}\right]$, respectively. We obtain smooth versions of these estimators by applying a smoother onto the two sets of estimated values $\{(t_p, t_{p'}), \widetilde{E}_V(t_p, t_{p'}), p, p' = 1, \ldots, m \text{ and } p \neq p'\}$ and $\{(t_p, t_{p'}), \widetilde{E}_W(t_p, t_{p'}), p, p' = 1, \ldots, m \text{ and } p \neq p'\}$. We denote the smoothed estimators as $\widehat{E}_V(t,t')$ and $\widehat{E}_W(t,t')$ and use them to estimate the two covariate functions,

$$\widehat{\Sigma}_Z(t,t') = \frac{1}{g'\{\widehat{\mu}(t)\}g'\{\widehat{\mu}(t')\}}\left[\widehat{E}_V(t,t') - g\{\widehat{\mu}(t)\}g\{\widehat{\mu}(t')\}\right] \tag{12}$$

$$\widehat{\Sigma}_W(t,t') = \frac{1}{g'\{\widehat{\mu}(t)\}g'\{\widehat{\mu}(t')\}}\left(\widehat{E}_W(t,t')\right). \tag{13}$$

To ensure the estimates of the covariance functions are symmetric and semi-positive definite, we replace any negative eigenvalues in their spectral decompositions with zero and force symmetry. These estimates for the covariance functions are used to obtain the two sets of eigenfunctions and eigenvalues, where $\{\widehat{\phi}_{V,k}(\cdot), \widehat{\lambda}_{V,k}\}_{k\geq 1}$, $\{\widehat{\phi}_{W,k}(\cdot), \widehat{\lambda}_{W,k}\}_{k\geq 1}$ are the estimated eigenfunctions and eigenvalues of the respective covariance functions.

3.2.2 Rare Events: Exponential Approximation

In some datasets, there may be cases of rare events and the latent probability $g^{-1}\{\mu(t)\}$ is close to the end points of 0 or 1. For the remainder of the derivation, we assume that

1 is the rare event and $g^{-1}\{\mu(t)\}$ close to 0. The exponential approximation assumes that $\exp(x)$ is small enough, such that $\exp(x)/(1+\exp(x)) \approx \exp(x)(1-\exp(x))$ is a valid approximation. This approximation is incorporated in the Taylor series expansion. This method is presented and studied for the *gmFPCA* method in Serban et al. (2013). We focus on the *gmFPCA+gAR* scenario when there can be periods of inactivity. Applying the exponential approximation, we obtain

$$E\left\{X_{ij}(t)|S_{ij}=1\right\} \quad \approx \quad \exp\left\{\mu(t)+\Sigma_Z(t,t)/2\right\} - \exp\left\{2\mu(t)+2\Sigma_Z(t,t)\right\},$$

where $\Sigma_Z(t,t') = Cov\{Z(t), Z(t')\}$ and $\mu(\cdot)$ is the unconditional mean function. This approximation of the conditional expectation is accurate but difficult to estimate in practice. Thus we further assume that $\exp\{2\mu(t)+2\Sigma_Z(t,t)\} << \exp\{\mu(t)+\Sigma_Z(t,t)/2\}$ and obtain the approximation

$$E(X_{ij}(t)|S_{ij}=1) \approx \exp\{\mu(t)+\Sigma_V(t,t)/2+\Sigma_W(t,t)/2\}, \tag{14}$$

where $\Sigma_Z(t,t') = \Sigma_V(t,t') + \Sigma_W(t,t')$.

Using this approximation, estimates of the two covariance functions are obtained through the conditional expectations of the products,

$$E\{X_{ij}(t)X_{ij'}(t')=1|S_{ij}S_{ij'}=1\} \quad \approx \quad b(t)b(t')\exp\{\Sigma_V(t,t')\}$$

where $j \neq j\prime$ and $b(t)$ is used to denote the quantity $\exp\{\mu(t)+\Sigma_V(t,t)/2+\Sigma_W(t,t)/2\}$, for notational purposes and out of brevity. Similarly, the joint conditional expectation of individuals from the same time period $j = j'$ is approximated as

$$E\{X_{ij}(t)X_{ij}(t')=1|S_{ij}=1\} \quad \approx \quad b(t)b(t')\exp\{\Sigma_V(t,t')+\Sigma_W(t,t')\}.$$

Using these two approximations, estimates of the mean function $\mu(\cdot)$, and two covariance functions, $\Sigma_V(\cdot,\cdot)$ and $\Sigma_W(\cdot,\cdot)$, are obtained in following steps. First, we implement a working assumption of independence across i and j. Then, apply method of moment estimators of the obtain the smoothed estimations of the conditional expectations. For all $t_p \neq t_{p'}$, these estimators are defined as

$$\begin{aligned}
\widetilde{E}_T(t_p,t_{p'}) &= \frac{\sum_{i=1}^n \sum_{j=1}^{J_i-1} X_{ijp}X_{ijp'}\widetilde{S}_{ij}}{\sum_{i=1}^n \sum_{j=1}^{J_i-1} \widetilde{S}_{ij}}, \\
\widetilde{E}_V(t_p,t_{p'}) &= \frac{\sum_{i=1}^n \sum_{j=1}^{J_i-1} \sum_{j'>j} X_{ijp}X_{ij'p'}\widetilde{S}_{ij}\widetilde{S}_{ij'}}{\sum_{i=1}^n \sum_{j=1}^{J_i-1} \sum_{j'>j} \widetilde{S}_{ij}\widetilde{S}_{ij'}}
\end{aligned}$$

where $\widetilde{E}_T(t,t')$ and $\widetilde{E}_V(t,t')$ are the unsmoothed estimators of $E\{X_{ij}(t)X_{ij}(t') = 1|S_{ij}=1\}$ and $E\{X_{ij}(t)X_{ij'}(t')=1|S_{ij}=1\}$, respectively. We recover smooth versions of the estimators by applying a bivariate smoother onto the two sets of estimated values $\{(t_p,t_{p'}), \widetilde{E}_V(t_p,t_{p\prime}), p,p'=1,\ldots,m$ and $p \neq p'\}$ and $\{(t_p,t_{p'}), \widetilde{E}_T(t_p,t_{p\prime}), p,p' = 1,\ldots,m$ and $p \neq p'\}$. We denote the smoothed estimators as $\widehat{E}_T(t,t')$ and $\widehat{E}_V(t,t')$.

Recall that $b(\cdot)$ is an approximation of the conditional expectation $E(X_{ij}(t)|S_{ij}=1)$. An unsmoothed estimator of $b(\cdot)$, is obtained by evaluating the estimator $\widetilde{b}(t)$ at each observed t, where

$$\widetilde{b}(t_p) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{J_i} X_{ijp}\widetilde{S}_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{J_i}\widetilde{S}_{ij}}.$$

A smooth estimator of $b(\cdot)$ is obtained by applying a smoother onto estimated values $\{t_p : \widetilde{b}(t_p), p = 1,\ldots,m\}$ and we denote this smooth estimator as $\widehat{b}(\cdot)$. Using these smooth estimates, we estimate the covariance functions $\Sigma_V(t,t')$ and $\Sigma_W(t,t')$ as

$$\widehat{\Sigma}_V(t,t') = \log\left\{\frac{\widehat{E}_V(t,t')}{\widehat{b}(t)\widehat{b}(t')}\right\} \tag{15}$$

$$\widehat{\Sigma}_W(t,t') = \log\left\{\frac{\widehat{E}_T(t,t')}{\widehat{b}(t)\widehat{b}(t')}\right\} - \widehat{\Sigma}_V(t,t'). \tag{16}$$

In order to ensure valid covariance functions, the negative eigenvalues are dropped in the spectral decomposition. As outlined in 2.3, through a spectral decomposition of these estimated covariance functions, we can estimate the first and second level eigenfunctions and their corresponding eigenvalues, $\left\{\widehat{\phi}_{V,k},\widehat{\lambda}_{V,k}\right\}_{k\geq 1}$ and $\left\{\widehat{\phi}_{W,k},\widehat{\lambda}_{W,k}\right\}_{k\geq 1}$. The number of eigenfunctions for the finite truncations, K_V and K_W, are selected by proportion of variation explained (PVE).

The mean function is estimated by plugging in the estimators into equation 14 and solving for $\mu(t)$,

$$\widehat{\mu}(t) = \left[\log\{\widehat{b}(t)\} - \widehat{\Sigma}_V(t,t)/2 - \widehat{\Sigma}_W(t,t)/2\right].$$

We have now obtained the estimates for the mean function and sets of eigenfunctions for the rare event scenario through the exponential approximation. As previous noted the derivations for the *gmFPCA* models are easily obtained from these derivations. These estimated values of $\widehat{\mu}(\cdot)$, $\left\{\widehat{\phi}_{V,k},\widehat{\lambda}_{V,k}\right\}_{k=1}^{K_V}$, and $\left\{\widehat{\phi}_{W,k},\widehat{\lambda}_{W,k}\right\}_{k=1}^{K_W}$ are used in the GLMM to estimate the first and second level coefficients. Using these estimated coefficients, classification rules are developed following the procedures outlined in Section 2.3 and further described in Weishampel et al. (2021b).

4 INTRODUCTION TO GFPCACLASSIF

In this section, we outline the functions of the **gFPCAClassif** package which are used to fit the three models of Section 2. The arguments of these functions are explained. We also show the fitted models are used to predict the classes of new data.

4.1 Fitting the Models

The **gFPCAClassif** package is built around two main functions which are used to fit the models presented in Section 2: *gsFPCA()* and *gmFPCA()*. The *gsFPCA()* function corresponds to fitting the *gsFPCA* model from Section 2.1 and the *gmFPCA()* function corresponds to fitting the two models from Sections 2.2-2.3, where the argument *gAR* is used within the *gmFPCA()* function to distinguish the two models.

We begin by presenting the arguments used in the two functions used to fit the different models. First, we examine the arguments in the function *gsFPCA()*:

```
gsFPCA(X_dat_s, Ys, covariates = NA, pve = 0.95,
        Kb = 10, num_knots = 10, bs0 = "cr").
```

The function estimates the various features of the gsFPCA model and trains the classifiers for binary data. The first required argument for the function is the *X_dat_s*, which is the formatted binary data. The binary data presented as matrix of (n, m) dimensions with n binary series observed at m time points. As noted earlier, we assume that the data for each individual is equispaced and regular. Therefore the bianry data for each individual will contain the same number of observations m. The second required argument is *Ys*, which is the group labels for each individual. The labeled are formatted as a vector of n elements with each element corresponding to the n rows of binary data. In the scenarios when additional covariates are available for the data, they can be included in the classifier using the *covariates* argument. The additional covariates used for the Bayes classifier need to be presented as a data frame of with n rows and the Q covariate variables, where each row corresponds to the n rows of binary data.

The remaining arguments contain the tuning parameters used to fit the model. The first of these remaining argument is the set of known basis functions. These basis functions are used to initially estimate the smooth latent trajectories for each individual. The number and type of basis function's is determined by Kb and $bs0$, respectively. The different types of basis functions are provided further described in the *mgcv* package (Wood, 2021). The mean function and eigenfunctions are estimated by a standard functional principal component analysis on the estimated smooth curves (Ramsay & Silverman, 2005; Xiao et al., 2013), with the number of components being determined by *PVE*.

The second function is used to fit the *gmFPCA* and *gmFPCA+gAR* models and is defined with the following arguments:

```
gmFPCA(X_dat_m, Ys, J, N, covariates = NA, pve1 = 0.95,
        pve2 = 0.95, approximation = "linear", bs0 = "cr",
        Kb = 5, gAR = FALSE, gar_covariates = NA, q = 3).
```

The group labels and the covariates in the model, *Ys* and *covariates* are defined in the same manner as the arguments in the *gsFPCA()* function. The binary data for the multi-level models is set by *X_dat_m*, where the binary data is formatted as a matrix of (nJ, m) dimensions with nJ binary series observed at m time points. The number of series per individual and the number of individuals are set by arguments J and N,

respectively. The type of approximation (Section 3) used to estimate the mean function and the eigenfunctions are determined by the argument *approximation*, where it take values of "linear" or "exponential". The number of functions and type of functions used in the basis when smoothing the covariance function estimates are set by arguments *Kb* and *bs0*. The number of eigenfunctions K_V and K_W are controlled by arguments *pve1* and *pve2*, where *pve1* and *pve2* are the PVE values for the individual level signal and within individual level signal, respectively.

Both the *gmFPCA* and the *gmFPCA+gAR* models are fitted using the *gmFPCA()* function. The two models are differentiated by the argument *gAR*, when this argument is set to *FALSE* the *gmFPCA* is trained using the data. Conversely when *gAR* is set to be *TRUE*, then the *gmFPCA+gAR* is trained. For the *gAR* models, the number of lags in the model is set by q. Additional covariates in the *gAR* model, are included with the *gar_covariates* argument, where *gar_covariates* is a data frame with n rows and all of the covariates to be included in the *gAR*.

These functions return the mean function, eigenfunctions, and principal component scores for each individual. For the *gmFPCA+gAR* models, the function also returns the fitted gAR model too. The output of these functions are then used to predict the groups of new individuals.

4.2 Predicting New Groups

For both of the models, two corresponding functions are included in the library to predict the group labels for new individuals. After the models are estimated using the previous functions, the prediction functions predict the group of a new individual given the individual's binary data and covariates values. The two prediction functions *gsFPCA_predict()* and *gmFPCA_predict()* are used for the *gsFPCA* and *gmFPCA* models, respectively. For the *gsFPCA* model, the predict function and its arguments are presented as,

```
gsFPCA_predict(gsFPCA.model, X_dat_s_new,
        covariates_new = NA),
```

where *gsFPCA.model* is the trained *gsFPCA* model, *X_dat_s_new* is the new binary data, and *covariates_new* is the covariate data for the new individuals. The new binary data, *X_dat_s_new*, is a matrix with m columns and number of rows equal to the number of new individuals. Similarly the *covariates_new* is a data frame with Q variables and number of rows equal to the number of new individuals.

Similarly, to predict the groups for new individuals when there are multiple series per individual, the prediction function and its arguments is defined as

```
gmFPCA_predict(gmFPCA.model, X_dat_m_new, covariates_new = NA,
        gar_covariates_new = NA),
```

where *gmFPCA.model* is the trained *gmFPCA* model, *X_dat_m_new* is the new binary data, and *covariates_new* is the data frame of covariates for the new individuals. The new binary data are formatted as a matrix containing J rows per each new individual and m columns for the observed time points. For *gmFPCA+gAR* models with covariates in the gAR model,

the gAR covariates for new individuals are included with the *gar_covariates_new* argument. Similar to the other covariates, the *gar_covariates_new* is a data frame containing a row for each new individual and variables corresponding to the gAR variables. Both prediction functions return the predicted group for each one of the new individuals, where the predictions are based on the methods outlined in Section 2.2.

5 APPLICATION IN SOCIAL MEDIA

To demonstrate the **gFPCAClassif** library and its functions, we apply the methods to a sample dataset of social media data. In this social media example, the goal is to identify the automated accounts from the genuine accounts based on their posting behaviors and additional covariates. The dataset presented is a subset of accounts studied in Cresci et al. (2017a), Weishampel et al. (2021a), and Weishampel et al. (2021b).

Before we are able to analyze the sample data, the package needs to be loaded.

```
R> library(gFPCAClassif)
```

The provided social media dataset contain data on 500 accounts, where the accounts are divided into a training set of 400 accounts and a testing set of 100 accounts. The identification numbers have been altered from the original datasets to conceal the accounts' identities. For each one of these accounts, the dataset contain the posting data of these accounts during a two week period summarized as binary data and additional covariates about the accounts.

For each account, the posting behavior over a two week period is summarized by binning the two weeks into 30 minute non-overlapping continuous time intervals and recording the posting activity of the account within the time interval. The data is recorded as 0 if the account did not post during the specified time interval and 1 if the account did post during the respective interval. The resulting binary data of the accounts are provided in matrices *X_dat_train* and *X_dat_test*, where the former contains the data for the training set accounts and the latter contains the data of the accounts in testing set. Additionally, we are provided with a few additional data about the accounts in the data frames *acc_data_train* and *acc_data_test*.

```
R> head(acc_data_train)
```

```
 id   followers_count friends_count created_at  group
 83         3834           3850         1/17/2012      2
484            0              0         1/16/2012      2
362         3152           2522        11/20/2008      1
258         4421           4827        11/22/2011      2
116         1846            776        12/15/2010      1
387            1              0         3/28/2012      2
```

For each individual (id), we have the group of the account (group) bot (group = 2) and genuine (group = 1). Additionally for each individual, we have the number of accounts which the individual follows (friends_count), the number of accounts which follow the individual (followers_count), and the date when the account was created (created_at).

5.1 gsFPCA()

To model the binary data, we fit the *gsFPCA* model without covariates, the number of eigenfucntions are determined by a PVE of 0.95,

```
R> gsFPCA.model = gsFPCA(X_dat_s = X_dat_train,
                Ys = acc_data_train$group, covariates = NA,
                pve = 0.95, Kb = 10, bs0="cr")
```

The smooth trajectories are estimated using the 10 cubic regression splines. The resulting object, *gsFPCA.model*, contains the estimated mean function, eigenfunctions, and the account's corresponding principal component scores. The estimated principal components scores for each account can be easily viewed in the output by:

```
R>  head(gsFPCA.model$scores_train)

            psi1        psi2         psi3
[1,]  0.46425138  -0.23886527  -0.12459635
[2,]  0.68930728  -0.12412110  -0.19138972
[3,] -0.57787830  -0.26004744   0.03867028
[4,]  0.73441177  -0.21517720  -0.16258036
[5,] -0.01012554  -0.06334019  -0.08914326
[6,]  0.34669469  -0.18249006   0.05152066
```

By providing the eigenfuntions and scores, researchers are able to interpret the values of the principal component scores and better understand the accounts' posting patterns. The eigenfunctions are displayed Figure 1 and obtained by using the *matplot()* function:

```
R>  matplot(gsFPCA.model$eigen_funcs, type="l", lwd = 2,
        lty = 1, xlab = "Day",
        main = "gsFPCA Eigenfunctions", xaxt="n")
R>  axis(1, at = (0:13)*48+1, labels = 1:14)
R>  legend("topright",
       legend = 1:length(colnames(gsFPCA.model$scores_train)),
       col = 1:length(colnames(gsFPCA.model$scores_train)),
       lty=1, lwd=2)
```

The eigenfunctions in Figure 1 explain the variability in posting patterns. For example the first eigenfunction has a slight decline over the two weeks and is positive throughout the study duration. This eigenfunction indicates that the overall frequency of posts is a main source of variability among accounts' posting patterns. The second eigenfunction addresses differences in the posting behavior between the first week versus the second week. Finally, the third eigenfunction addresses differences of the posting habits during the middle of the study from the posting habits at the beginning and end of the study.

To predict the groups of new individuals we use the *gsFPCA_predict()* function and input the trained *gsFPCA* with the posting data from the new accounts,

```
R>  gsFPCA.results = gsFPCA_predict(gsFPCA.model,
                        X_dat_s_new = X_dat_test)
```

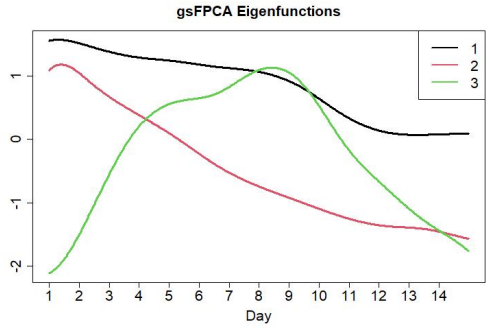

Figure 1. Estimated eigenfunctions for the social media data using the *gsFPCA()* functions. The eigenfunctions are defined over the two week time period of the data.

The *gsFPCA_predict()* function predicts the groups of new accounts based on their binary functional data. Comparing the results to true values show us that the *gsFPCA* accurately discriminate 88% of the users.

```
R>  table(gsFPCA.results, acc_data_test$group)

gsFPCA.results  1  2
             1 44  6
             2  6 44
```

The previous analysis trained and tested the classifier using only the binary-valued functional data. We can include the number of followers for each account as an additional covariate in the nonparametric Bayes classifier. The updated model was fitted considering the number of followers and binary functional data is fit and evaluated by:

```
R>  gsFPCA.model.ff = gsFPCA(X_dat_s = X_dat_train,
                      Ys = acc_data_train$group,
                      covariates = cbind.data.frame(followers =
                      acc_data_train$followers_count),
                      pve = 0.95, Kb = 10, bs0="cr")
R>  gsFPCA.results = gsFPCA_predict(gsFPCA.model.ff,
                      X_dat_s_new =  X_dat_test,
                      covariates_new=cbind.data.frame(followers=
                      acc_data_test$followers_count))
R>  table(gsFPCA.results, acc_data_test$group)

gsFPCA.results  1  2
             1 46 18
             2  4 32
```

Adding the number of followers for each account actually decreased the performance of the classifier as the updated classifier accurately labeled 78% of the new accounts. Similar

results are observed when including the number of friends into the models. These findings are in agreement with the findings of Chavoshi et al. (2016). Automated accounts are able to mimic many features of genuine accounts, but the posting behaviors remains one of the main indicators of inauthentic accounts (Chavoshi et al., 2016, 2017; Cresci et al., 2017b).

5.2 gmFPCA()

As explained in Section 4, Both the *gmFPCA* and the *gmFPCA+gAR* models are implemented using the *gmFPCA()* function. To apply the multi-level structure to the social media data, the provided binary data need to be restructured. In the previous approach, the binary data for an account was treated as a single series of observations from the two week interval. In the multi-level approach, the data consists multiple series per individual. In the social media application we consider each day is a separate series. Due to the nature of the social media data, there are days, when the account does not post at any point during day. Because there can be inactive days the *gmFPCA+gAR* is more appropriate to model the data.

Recall, the binary data are formatted based on 30 minute intervals therefore there are 48 observations per day, $m = 48$. Therefore to format the binary-valued matrix as described in Section 4, we need to have each row correspond to a given individual on a given day. The formatted binary data of the training and testing set accounts for the multi-level models are provided in the *X_dat_m_train* and *X_dat_m_test* matrices. The formatted data is used to fit the *gmFpCA+gAR* by:

```
R>  gmfpca.cur = gMFPCA(X_dat = X_dat_m_train,
                    Ys = acc_data_train$group, J = 14, N=400,
                    covariates = NA, gAR = T, q = 3, pve1 = 0.95,
                    pve2 = 0.75,  approximation = "linear")
```

Based on the data, the probability of an account posting within the 30 minutes interval is large enough such that the linear approximation method is recommended. If this probability is small and posts are rare, then the exponential approximation is more appropriate. The trained *gmfpca* model contains the mean function, both sets of eigenfunctions, scores, and the trained *gAR* models for each group. The individual level eigenfunctions (Figure 2) can be displayed using the following code. The individual level eigenfuntions explain how the posting patterns within a day differs among accounts.

```
R>  matplot(gmfpca.cur$eigen_funcs1, type="l", lwd = 3, lty = 1,
        xlab = "Time", main = "gsFPCA Eigenfunctions", xaxt="n")
R>  axis(1, at = (0:4)*12, labels = c("12am", "6am", "12pm",
        "6pm", "12am"))
R>  legend("bottomright",
        legend=c(1:length(gmfpca.cur$eigen_vals1)),
        col=1:length(gmfpca.cur$eigen_vals1), lty=1, lwd=2)
```

Because there are two groups in the provided data, there are two fitted *gAR* models in the *gmfpca.cur* object, one *gAR* model for each group. The lag in the both of these *gAR* models is to be $q = 3$. The fitted *gAR* model for the genuine accounts can be retrieved and viewed by

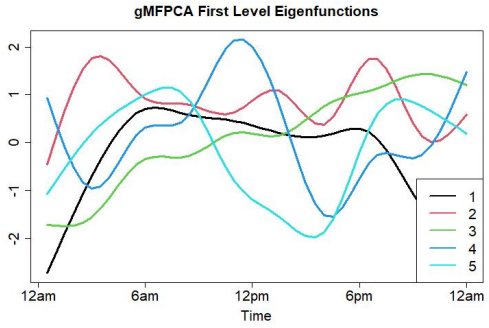

Figure 2. Estimated first level eigenfunctions for the social media data using the *gmFPCA()* function. The eigenfunctions are defined throughout the day.

```
R>   gmfpca.cur$gar_models_ls[[2]]

$model

Call:  bayesglm(formula = formula.cur, family = binomial,
    data = data.s.cur2, prior.df = Inf, scaled = FALSE)

Coefficients:
(Intercept)           S1           S2           S3
    -0.9244       2.8785       2.2737       0.4148

Degrees of Freedom: 2199 Total (i.e. Null);  2196 Residual
Null Deviance:     367.5
Residual Deviance: 279.2  AIC: 287.2

$initial_probs
[1] 0.005415162 0.001805054 0.001353791 0.010379061 0.002707581
0.009927798 0.012184116 0.956227437
```

The model consists of two components: the autoregressive model estimates and the initial probabilities. The initial probabilities are the probabilities which define the estimated probability mass function for the possible combinations of the first q initial states. From the fitted *gAR* it is clear that the automated account's active state is highly affected by the active status of the previous two days.

The *gmFPCA_predict()* function is used to predict the groups given the binary-valued functional data for new accounts.

```
R>  gmFPCA.results = gmFPCA_predict(gmfpca.cur,
                                X_dat_m_new = X_dat_m_test)

R>  table(gmFPCA.results, acc_data_test$group)
```

```
gmFPCA.results  1  2
             1 50  1
             2  0 49
```

The trained *gmFPCA+gAR* classifier is able to accurately discriminate 99% of the accounts in the testing set. This accuracy is larger than the gsFPCA approach, suggesting that group differences occur in the within day posting patterns.

6 CONCLUSION

The **gFPCAClassif** package presented in this paper provides software analyze and classify of binary-valued functional data. One major contribution of this package is that it provides methods to model and classify binary-valued functional data. The package was developed around three different model-based classifiers: *gsFPCA*, *gmFPCA*, and *gmFPCA + gAR*. Each one of the models addresses different data structures and dependencies. The package contains methods to fit these models, train nonparametric Bayesian classifiers, and predict classes for new data based upon the previously discussed methods. The theory and estimation procedures used for fitting these models were briefly presented. The functions and the arguments used to fit the models were discussed. This package is the first software to evaluate the presented methods. As show in the application, the proposed methods can be implemented and effective on real data. Future versions of the packages will account for scenarios when the data is not observed at points which are equispaced and regular. This expansion will allow the researcher to use these methods to less frequently recorded data. As methods are developed to account for additional correlation structures of the binary-valued functional data and expanded for categorical-valued functional data, they will be incorporated into the package.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the NSF MMS grant number: SES 2020179.

REFERENCES

Bande, M. F., de la Fuente, M. O., Galeano, P., Nieto, A., Garcia-Portugues, E., & de la Fuente, M. M. O. (2020). *fda.usc: Functional Data Analysis and Utilities for Statistical Computing*. R package version 2.0.2.
URL `http://CRAN.R-project.org/package=fda.usc`

Breslow, N. E., & Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American statistical Association*, *88*(421), 9–25.

Chavoshi, N., Hamooni, H., & Mueen, A. (2016). Debot: Twitter bot detection via warped correlation. In *16th International Conference on Data Mining*, (pp. 817–822).

Chavoshi, N., Hamooni, H., & Mueen, A. (2017). Temporal patterns in bot activities. In *Proceedings of the 26th International Conference on World Wide Web Companion*, (pp. 1601–1606).

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2017a). The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th International Conference on World Wide Web Companion*, (pp. 963–972).

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2017b). Social fingerprinting: detection of spambot groups through DNA-inspired behavioral modeling. *IEEE Transactions on Dependable and Secure Computing*, *15*(4), 561–576.

Dai, X., Müller, H.-G., & Yao, F. (2017). Optimal Bayes classifiers for functional data and density ratios. *Biometrika*, *104*(3), 545–560.

Delaigle, A., & Hall, P. (2012). Achieving near perfect classification for functional data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *74*(2), 267–286.

Di, C.-Z., Crainiceanu, C., Caffo, B., & Punjabi, N. (2009). Multilevel functional principal component analysis. *The Annals of Applied Statistics*, *3*(1), 458–488.

Diggle, P., Diggle, P. J., Heagerty, P., Liang, K.-Y., Heagerty, P. J., Zeger, S., et al. (2002). *Analysis of longitudinal data*. Oxford university press.

Ferraty, F., & Vieu, P. (2006). *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media.

Gajardo, A., Carroll, C., Chen, Y., Dai, X., Fan, J., Hadjipantelis, P. Z., Han, K., Ji, H., Mueller, H.-G., & Wang, J.-L. (2021). *fdapace: Functional Data Analysis and Empirical Dynamics*. R package version 0.5.7.
URL `https://CRAN.R-project.org/package=fdapace`

Goldsmith, J., Scheipl, F., Huang, L., Wrobel, J., Gellar, J., Harezlak, J., McLean, M., Swihart, B., Xiao, L., Crainiceanu, C., et al. (2020). *Package 'Refund'*.
URL `http://CRAN.R-project.org/package=refund`

Goldsmith, J., Zipunnikov, V., & Schrack, J. (2015). Generalized multilevel function-on-scalar regression and principal component analysis. *Biometrics*, *71*(2), 344–353.

Hall, P., Müller, H.-G., & Yao, F. (2008). Modelling sparse generalized longitudinal observations with latent gaussian processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*(4), 703–723.

Hall, P., Poskitt, D. S., & Presnell, B. (2001). A functional data analytic approach to signal discrimination. *Technometrics*, *43*(1), 1–9.

Heagerty, P. J. (1999). Marginally specified logistic-normal models for longitudinal binary data. *Biometrics*, *55*(3), 688–698.

Hsu, C.-C., Huang, Y.-P., & Chang, K.-W. (2008). Extended naive bayes classifier for mixed data. *Expert Systems with Applications*, *35*(3), 1080–1083.

Huang, W., & Ruppert, D. (2019). Copula-based functional Bayes classification with principal components and partial least squares. *arXiv preprint arXiv:1906.00538*.

Johnson, D. S., Conn, P. B., Hooten, M. B., Ray, J. C., & Pond, B. A. (2013). Spatial occupancy models for large data sets. *Ecology*, *94*(4), 801–808.

Liang, K.-Y., & Zeger, S. L. (1986). Longitudinal data analysis using generalized linear

models. *Biometrika*, *73*(1), 13–22.

Lipsitz, S. R., Laird, N. M., & Harrington, D. P. (1991). Generalized estimating equations for correlated binary data: using the odds ratio as a measure of association. *Biometrika*, *78*(1), 153–160.

McLean, M. W., Hooker, G., Staicu, A.-M., Scheipl, F., & Ruppert, D. (2014). Functional generalized additive models. *Journal of Computational and Graphical Statistics*, *23*(1), 249–269.

Overgoor, G., Chica, M., Rand, W., & Weishampel, A. (2019). Letting the computers take over: Using ai to solve marketing problems. *California Management Review*, *61*(4), 156–185.

Peng, J., & Paul, D. (2011). *fpca: Restricted MLE for Functional Principal Components Analysis*. R package version 3.5.
URL `http://CRAN.R-project.org/package=fpca`

Ramsay, J., & Silverman, B. W. (2005). *Functional Data Analysis*. Springer.

Ramsay, J., Wickham, H., Ramsay, M. J., & deSolve, S. (2012). *fda: Functional Data Analysis*. R package version 2.2.8.
URL `http://CRAN.R-project.org/package=fda`

Rao Chaganty, N., & Joe, H. (2004). Efficiency of generalized estimating equations for binary responses. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *66*(4), 851–860.

Scheipl, F., Gertheiss, J., Greven, S., et al. (2016). Generalized functional additive mixed models. *Electronic Journal of Statistics*, *10*(1), 1455–1492.

Serban, N., Staicu, A.-M., & Carroll, R. J. (2013). Multilevel cross-dependent binary longitudinal data. *Biometrics*, *69*(4), 903–913.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, vol. 26. CRC press.

Weishampel, A., Staicu, A.-M., & Rand, W. (2021a). Classification of social media users using a generalized functional analysis.

Weishampel, A., Staicu, A.-M., & Rand, W. (2021b). Classification of social media users using generalized multilevel functional analysis.

Wood, S. (2021). *Package 'mgcv'*.
URL `http://CRAN.R-project.org/package=mgcv`

Wood, S. N., Pya, N., & Säfken, B. (2016). Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association*, *111*(516), 1548–1563.

Xiao, L., Li, Y., & Ruppert, D. (2013). Fast bivariate P-splines: the sandwich smoother. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *75*(3), 577–599.

Yao, F., Müller, H.-G., & Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, *100*(470), 577–590.

Ye, J., Li, Y., Guan, Y., et al. (2015). Joint modeling of longitudinal drug using pattern

and time to first relapse in cocaine dependence treatment data. *The Annals of Applied Statistics*, *9*(3), 1621–1642.

6.0.1 Non-rare Events: Linear Approximation

When fitting the gmFPCA models and assuming that the probability of $P(X_{ijp} = 1|t_p)$ is not extreme, we estimate the mean and covariance functions, by using a linear approximation of the link function (Hall et al., 2008; Serban et al., 2013). Assuming that the variation of $Z_{ij}(\cdot)$ around its mean is small, we approximate $g\left\{Z_{ij}(t)\right\}$ via a Taylor expansion (Serban et al., 2013)

$$\begin{aligned} g\left\{Z_{ij}(t)\right\} \quad \approx \quad & g\left\{\mu(t)\right\} + g'\left\{\mu(t)\right\}\left\{V_i(t) + W_{ij}(t)\right\} + \\ & \frac{1}{2} g''\left\{\mu(t)\right\}\left\{V_i(t) + W_{ij}(t)\right\}^2, \end{aligned}$$

where $g'(t) = \partial g(t)/\partial t$ and $g''(t) = \partial^2 g(t)/\partial t^2$. Omitting the term with the second derivative of provides a linear approximation for the marginal probability (Serban et al., 2013). By only including the first two terms of the approximation we obtain $Pr\{X_{ij}(t) = 1\} \approx g\{(\mu(t)\}$. Using this approximation the mean function is approximated by first estimating the probability function $\widehat{p}(t)$ through a univariate smoother onto $\left\{\left(t_p, (\sum_{i=1}^n J_i)^{-1} \sum_{i=1}^n \sum_{j=1}^J X_{ijp}\right) : p = 1, \ldots, m\right\}$ and then $\widehat{\mu}(t) = g^{-1}\{\widehat{p}(t)\}$. This estimate is obtained by implementing a working assumption of independence across i and j.

To estimate the various covariance functions and the eigenfunctions, the linear approximation is applied to the marginal joint probability,

$$\begin{aligned} Pr\left(X_{ijp}X_{ij'p'} = 1|t_p, t_{p'}\right) \approx & g\{\mu(t_p)\}g\{\mu(t_{p'})\} + \\ & \left[Cov(V_i\{t_p), V_i(t_{p'})\} + Cov\{W_{ij}(t_p), W_{ij'}(t_{p'})\}\mathbb{1}_{(j=j')}\right] \\ & \times g'\{\mu(t_p)\}g'\{\mu(t_{p'})\} \end{aligned}$$

where $\mathbb{1}_{(j=j')}$ is the indicator function, for $j = j'$. Continuing to use the working assumption of independence independence across i and j, estimates of the two covariance functions, $\Sigma_V(\cdot,\cdot)$ and $\Sigma_W(\cdot,\cdot)$, are obtained in following steps. For all $t_p \neq t_{p'}$ denote

$$\widetilde{E}_W(t_p, t_{p\prime}) = \frac{\sum_{i=1}^n \sum_{j=1}^{J_i - 1} \sum_{j'>j} \left(X_{ijp} - X_{ij'p}\right)\left(X_{ijp'} - X_{ij'p'}\right)}{\sum_{i=1}^n J_i(J_i - 1)}, \quad (17)$$

$$\widetilde{E}_V(t_p, t_{p\prime}) = \frac{\sum_{i=1}^n \sum_{j=1}^{J_i - 1} \sum_{j'>j} X_{ijp}X_{ij'p'}}{\sum_{i=1}^n J_i(J_i - 1)/2} \quad (18)$$

as unsmoothed estimators evaluated at all observed time points. We obtain smooth versions of these estimators by applying a smoother onto the two sets of estimated values $\{(t_p, t_{p'}), \widetilde{E}_V(t_p, t_{p'}), p, p' = 1, \ldots, m \text{ and } p \neq p'\}$ and $\{(t_p, t_{p'}), \widetilde{E}_W(t_p, t_{p'}), p, p' =$

$1, \ldots, m$ and $p \neq p'\}$. We denote the smoothed estimators as $\widehat{E}_V(t,t')$ and $\widehat{E}_W(t,t')$ and use them to estimate the two covariate functions,

$$\widehat{\Sigma}_Z(t,t') = \frac{1}{g'\{\widehat{\mu}(t)\}g'\{\widehat{\mu}(t')\}}\left[\widehat{E}_V(t,t') - g\{\widehat{\mu}(t)\}g\{\widehat{\mu}(t')\}\right] \quad (19)$$

$$\widehat{\Sigma}_W(t,t') = \frac{1}{g'\{\widehat{\mu}(t)\}g'\{\widehat{\mu}(t')\}}\left(\widehat{E}_W(t,t')\right). \quad (20)$$

To ensure the estimates of the covariance functions are symmetric and semi-positive definite, we replace any negative eigenvalues in their spectral decompositions with zero and force symmetry. These estimates for the covariance functions are used to obtain the two sets of eigenfunctions and eigenvalues, where $\{\widehat{\phi}_{V,k}(\cdot), \widehat{\lambda}_{V,k}\}_{k\geq 1}$, $\{\widehat{\phi}_{W,k}(\cdot), \widehat{\lambda}_{W,k}\}_{k\geq 1}$ are the estimated eigenfunctions and eigenvalues of the respective covariance functions.

6.0.2 gmFPCA+gAR

When fitting the both multi-level models and assuming that the probability of $P(X_{ijp} = 1|t_p)$ is not extreme, we estimate the mean and covariance functions by using a linear approximation of the link function (Hall et al., 2008; Serban et al., 2013). The derivation and estimation methods for the two multi-level models are similar. Altering the estimating methods to obtain the mean function and eigenfunctions from the *gmFPCA+gAR* model for the *gmFPCA* are trivial. This is because latent state of the *gmFPCA+gAR* can be considered as always being active to accommodate the *gmFPCA* model. Therefore we will only present the methods and derivations for the *gmFPCA+gAR*. The estimation procedures for the *gmFPCA* model is included in the supplemental material.

Unlike the models in Hall et al. (2008) and Serban et al. (2013), the proposed model contains an latent state indicator variable defining active days. In order to adapt the Taylor approximation to the proposed model, we use the expectation conditioning on $S_{ij} = 1$ and only consider days when the accounts are active online. It follows that $E\left[X_{ijp}|S_{ij} = 1, t_p\right] = E\left[g\left\{Z_{ij}(t_p)\right\}\right]$ and this quantity can be used to estimate $g\left\{\mu(t_p)\right\}$, and thus ultimately estimate the mean $\mu(t)$. Since the mean function is assumed to be continuous over the domain, the mean is estimated in two steps: first a univariate smoother is passed through the data to estimate the probability $\widehat{p}(t)$

$$\left\{\left(t_p, \frac{\sum_{i-1}^{n}\sum_{j=1}^{J_i}\widetilde{S}_{ij}X_{ijp}}{\sum_{i-1}^{n}\sum_{j=1}^{J_i}\widetilde{S}_{ij}}\right) : p = 1, \ldots, m\right\}$$

and then $\widehat{\mu}(t) = g^{-1}\left\{\widehat{p}(t)\right\}$.

Assuming that the variation of $Z_{ij}(\cdot)$ around its mean is small, we approximate $g\left\{Z_{ij}(t)\right\}$ via a Taylor expansion (Serban et al., 2013)

$$g\left\{Z_{ij}(t)\right\} \approx g\left\{\mu(t)\right\} + g'\left\{\mu(t)\right\}\left\{V_i(t) + W_{ij}(t)\right\} + \frac{1}{2}g''\left\{\mu(t)\right\}\left\{V_i(t) + W_{ij}(t)\right\}^2,$$

where $g'(t) = \partial g(t)/\partial t$ and $g''(t) = \partial^2 g(t)/\partial t^2$. Omitting the term with the second derivative of provides a linear approximation for the marginal probability (Serban et al., 2013). By only including the first two terms of the approximation we obtain $Pr\{X_{ij}(t) = 1\} \approx g\{(\mu(t)\}$. Using this approximation the mean function is approximated by first estimating the probability function $\widehat{p}(t)$ through a univariate smoother onto $\left\{\left(t_p, (\sum_{i=1}^n J_i)^{-1} \sum_{i=1}^n \sum_{j=1}^J X_{ijp}\right) : p = 1, \ldots, m\right\}$ and then $\widehat{\mu}(t) = g^{-1}\{\widehat{p}(t)\}$. This estimate is obtained by implementing a working assumption of independence across i and j.

To estimate the various covariance functions and the eigenfunctions, the linear approximation is applied to the marginal joint probability,

$$\begin{aligned} Pr\left(X_{ijp}X_{ij'p'} = 1 | t_p, t_{p'}\right) \approx & g\{\mu(t_p)\}g\{\mu(t_{p'})\} + \\ & \left[Cov(V_i\{t_p), V_i(t_{p'})\} + Cov\{W_{ij}(t_p), W_{ij'}(t_{p'})\}\mathbb{1}_{(j=j')}\right] \\ & \times g'\{\mu(t_p)\}g'\{\mu(t_{p'})\} \end{aligned}$$

where $\mathbb{1}_{(j=j')}$ is the indicator function, for $j = j'$. Continuing to use the working assumption of independence independence across i and j, estimates of the two covariance functions, $\Sigma_V(\cdot,\cdot)$ and $\Sigma_W(\cdot,\cdot)$, are obtained in following steps. For all $t_p \neq t_{p'}$ denote

$$\widetilde{E}_W(t_p, t_{p\prime}) = \frac{\sum_{i=1}^n \sum_{j=1}^{J_i - 1} \sum_{j'>j} \left(X_{ijp} - X_{ij'p}\right)\left(X_{ijp'} - X_{ij'p'}\right)}{\sum_{i=1}^n J_i(J_i - 1)}, \quad (21)$$

$$\widetilde{E}_V(t_p, t_{p\prime}) = \frac{\sum_{i=1}^n \sum_{j=1}^{J_i - 1} \sum_{j'>j} X_{ijp}X_{ij'p'}}{\sum_{i=1}^n J_i(J_i - 1)/2} \quad (22)$$

as unsmoothed estimators evaluated at all observed time points. We obtain smooth versions of these estimators by applying a smoother onto the two sets of estimated values $\{(t_p, t_{p'}), \widetilde{E}_V(t_p, t_{p'}), p, p' = 1, \ldots, m \text{ and } p \neq p'\}$ and $\{(t_p, t_{p'}), \widetilde{E}_W(t_p, t_{p'}), p, p' = 1, \ldots, m \text{ and } p \neq p'\}$. We denote the smoothed estimators as $\widehat{E}_V(t, t')$ and $\widehat{E}_W(t, t')$ and use them to estimate the two covariate functions,

$$\widehat{\Sigma}_Z(t, t') = \frac{1}{g'\{\widehat{\mu}(t)\}g'\{\widehat{\mu}(t')\}}\left[\widehat{E}_V(t, t') - g\{\widehat{\mu}(t)\}g\{\widehat{\mu}(t')\}\right] \quad (23)$$

$$\widehat{\Sigma}_W(t, t') = \frac{1}{g'\{\widehat{\mu}(t)\}g'\{\widehat{\mu}(t')\}}\left(\widehat{E}_W(t, t')\right). \quad (24)$$

To ensure the estimates of the covariance functions are symmetric and semi-positive definite, we replace any negative eigenvalues in their spectral decompositions with zero and force symmetry. These estimates for the covariance functions are used to obtain the two sets of eigenfunctions and eigenvalues, where $\{\widehat{\phi}_{V,k}(\cdot), \widehat{\lambda}_{V,k}\}_{k\geq 1}$, $\{\widehat{\phi}_{W,k}(\cdot), \widehat{\lambda}_{W,k}\}_{k\geq 1}$ are the estimated eigenfunctions and eigenvalues of the respective covariance functions.

6.1 Rare Events

In the previous section, we present methodology which estimates these features however these estimates run into some issues when the mean function $\mu(\cdot)$ is extreme, i.e. $g^{-1}\{\mu(\cdot)\}$ is close to 0 or 1. For these instances, Serban et al. (2013) presents methodology to estimate the mean functions and various eigenfunctions of the *gmFPCA* model, using an exponential approximation.

In these extreme datasets, there are few positive observations and the latent probability $g^{-1}(Z_{ij}(t))$ is closed to zero. The exponential approximation assumes that $\exp(x)$ is small enough, such that $\exp(x)/(1+\exp(x)) \approx \exp(x)(1-\exp(x))$. This approximation is incorporated in the Taylor series expansion. Since this method is presented and studied for the *gmFPCA* method in Serban et al. (2013), we will focus on the *gmFPCA+gAR* scenario when there can be periods of inactivity. Applying the exponential approximation to the Taylor series expansion, we obtain

Due to the estimation procedures of the single-level as outlined in Weishampel et al. (2021a), approach of estimating the latent trajectory using splines and the nature of single level data, the exponential approximation is only considered for the gmFPCA models.

6.1.1 Multi-level Models

In this model we assume that the binary series for a given i and j are independent realizations of the latent function $Z_{ij}(\cdot)$ and that $Z_{ij}(t)$ is approximated as

$$Z_{ij}(t) \approx \mu(t) + \sum_{k=1}^{K_v} \xi_{ik}\psi_{V,k}(t) + \sum_{k=1}^{K_W} \zeta_{ijk}\psi_{W,k}(t). \tag{25}$$

In some extreme datasets, there are few positive observations and the latent probability $g^{-1}(Z_{ij}(t))$ is closed to zero. The exponential approximation assumes that $\exp(x)$ is small enough, such that $\exp(x)/(1+\exp(x)) \approx \exp(x)(1-\exp(x))$ is a valid approximation. This approximation is incorporated in the Taylor series expansion. Since this method is presented and studied for the *gmFPCA* method in Serban et al. (2013), we will focus on the *gmFPCA+gAR* scenario when there can be periods of inactivity. Applying the exponential approximation to the Taylor series expansion, we obtain

$$E\left\{X_{ij}(t)|S_{ij}=1\right\} \quad \approx \quad \exp\left\{\mu(t)+\Sigma(t,t)/2\right\} - \exp\left\{2\mu(t)+2\Sigma(t,t)\right\},$$

where $\Sigma(t,t') = Cov\{Z(t), Z(t')\}$ and $\mu(\cdot)$ is the unconditional mean function. This approximation of the conditional expectation, $E(X(t)|S=1)$ is accurate but difficult to estimate. Thus we further assume that $\exp\{2\mu(t)+2\Sigma(t,t)\} << \exp\{\mu(t)+\Sigma(t,t)/2\}$ and approximate it as

$$E(X(t)|S=1) \approx \exp\{\mu(t)+\Sigma_V(t,t)/2+\Sigma_W(t,t)/2\}, \tag{26}$$

where $\Sigma(t,t') = \Sigma_V(t,t') + \Sigma_W(t,t')$.

Using this approximation, estimates of the two covariance functions are obtained through the conditional joint probabilities,

$$Pr\{X_{ij}(t)X_{ij'}(t') = 1 | S_{ij}S_{ij'} = 1\} \quad \approx \quad b(t)b(t')\exp\{\Sigma_V(t,t')\}$$

where $j \neq j\prime$ and $b(t)$ is used to denote the quantity $\exp\{\mu(t)+\Sigma_V(t,t)/2+\Sigma_W(t,t)/2\}$, for notational purposes and out of brevity. Similarly, the joint conditional probability of individuals from the same time period $j = j'$ is approximated as

$$Pr\{X_{ij}(t)X_{ij}(t') = 1 | S_{ij} = 1\} \quad \approx \quad b(t)b(t')\exp\{\Sigma_V(t,t') + \Sigma_W(t,t')\}.$$

Using these two approximations, estimates of the mean function $\mu(\cdot)$, and two covariance functions, $\Sigma_V(\cdot,\cdot)$ and $\Sigma_W(\cdot,\cdot)$, are obtained in following steps. First, we implement a working assumption of independence across i and j. Then, for all $t_p \neq t_{p'}$ denote by

$$\begin{aligned}
\widetilde{E}_W(t_p, t_{p\prime}) &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{J_i-1}\sum_{j'>j}\left(X_{ijp} - X_{ij'p}\right)\left(X_{ijp'} - X_{ij'p'}\right)\widetilde{S}_{ij}\widetilde{S}_{ij'}}{2\sum_{i=1}^{n}\sum_{j=1}^{J_i-1}\sum_{j'>j}\widetilde{S}_{ij}\widetilde{S}_{ij'}}, \\
\widetilde{E}_V(t_p, t_{p\prime}) &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{J_i-1}\sum_{j'>j}X_{ijp}X_{ij'p'}\widetilde{S}_{ij}\widetilde{S}_{ij'}}{\sum_{i=1}^{n}\sum_{j=1}^{J_i-1}\sum_{j'>j}\widetilde{S}_{ij}\widetilde{S}_{ij'}}
\end{aligned}$$

where $X_{ijp} = X_{ij}(t_p)$ and $\widetilde{S}_{ij} = \mathbb{1}_{\sum_{p=1}^{m} X_{ijp}>0}$. We estimate smooth versions of the estimators by applying a smoother onto the two sets of estimated values $\{(t_p, t_{p'}), \widetilde{E}_V(t_p, t_{p\prime}), p, p' = 1,\ldots,m \text{ and } p \neq p'\}$ and $\{(t_p, t_{p'}), \widetilde{E}_W(t_p, t_{p\prime}), p, p' = 1,\ldots,m \text{ and } p \neq p'\}$. We denote the smoothed estimators as $\widehat{E}_V(t, t\prime)$ and $\widehat{E}_W(t, t\prime)$. The number of eigenfunctions for the finite truncations, K_V and K_W, are selected by proportion of variation explained (PVE). In order to ensure semi-positive definite matrices, the negative eigenvalues are dropped in the spectral decomposition.

A smooth estimator of $b(\cdot)$, which as a reminder is the conditional expectation $E(X(t)|S = 1)$, is obtained by through evaluating for each observed t_p

$$\widetilde{b}(t_p) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{J_i}X_{ijp}\widetilde{S}_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{J_i}\widetilde{S}_{ij}}.$$

A smooth estimator of $b(\cdot)$ is obtained by applying a smoother onto estimated values $\{t_p : \widetilde{b}(t_p), p, p' = 1,\ldots,m\}$ and we denote this smooth estimator as $\widehat{b}(\cdot)$. Using these smooth estimates, we estimate the covariance functions $\Sigma_V(t,t')$ and $\Sigma_W(t,t')$ as

$$\widehat{\Sigma}_V(t,t') = \log\left(\frac{\widehat{E}_V(t,t')}{\widehat{b}(t)\widehat{b}(t')}\right) \tag{27}$$

$$\widehat{\Sigma}_W(t,t') = \log\left(\frac{\widehat{E}_W(t,t')}{\widehat{b}(t)\widehat{b}(t')}\right) - \widehat{\Sigma}_V(t,t'). \tag{28}$$

As outlined in 2.3, by decomposing these estimated covariance functions we can estimate the first and second level eigenfunctions and their corresponding eigenvalues, $\left\{\widehat{\phi}_{V,k}, \widehat{\lambda}_{V,k}\right\}_{k\geq 1}$ and $\left\{\widehat{\phi}_{W,k}, \widehat{\lambda}_{W,k}\right\}_{k\geq 1}$. Finally the mean function is estimated by plugging in the estimators into equation 14 and solving for $\mu(t)$,

$$\widehat{\mu}(t) \quad = \quad \left[\log(\widehat{b}(t)) - \widehat{\Sigma}_V(t,t)/2 - \widehat{\Sigma}_W(t,t)/2\right].$$

We have now obtained the estimates for the mean function and sets of eigenfunctions through the exponential approximation. The Classification rules are developed following the procedures outlined in Section 2.3 and further described in Weishampel et al. (2021b), using these new estimated features: $\widehat{\mu}(\cdot)$, $\left\{\widehat{\phi}_{V,k}, \widehat{\lambda}_{V,k}\right\}_{k=1}^{K_V}$, and $\left\{\widehat{\phi}_{W,k}, \widehat{\lambda}_{W,k}\right\}_{k=1}^{K_W}$.

6.1.2 gmFPCA+gAR

In this model, we assume that the binary data are observed independent realizations of the latent curve $Z_{ij}(t)$ and that $Z_{ij}(t)$ is approximated as

$$Z_{ij}(t) = \mu(t) + \sum_{k\geq 1} \xi_{ik}\psi_{V,k}(t) + \sum_{k\geq 1} \zeta_{ijk}\psi_{W,k}(t). \tag{29}$$

In some extreme datasets, there are few positive observations and the latent probability $g^{-1}(Z_{ij}(t))$ is closed to zero. The exponential approximation assumes that $\exp(x)$ is small enough, such that $\exp(x)/(1+\exp(x)) \approx \exp(x)(1-\exp(x))$ is a valid approximation. This approximation is incorporated in the Taylor series expansion. Since this method is presented and studied for the *gmFPCA* method in Serban et al. (2013), we will focus on the *gmFPCA+gAR* scenario when there can be periods of inactivity. Applying the exponential approximation to the Taylor series expansion, we obtain

$$E\left\{X_{ij}(t)|S_{ij}=1\right\} \quad \approx \quad \exp\left\{\mu(t)+\Sigma(t,t)/2\right\} - \exp\left\{2\mu(t)+2\Sigma(t,t)\right\},$$

where $\Sigma(t,t') = Cov\{Z(t), Z(t')\}$ and $\mu(\cdot)$ is the unconditional mean function. This approximation of the conditional expectation, $E(X(t)|S=1)$ is accurate but difficult to estimate. Thus we further assume that $\exp\{2\mu(t)+2\Sigma(t,t)\} << \exp\{\mu(t)+\Sigma(t,t)/2\}$ and approximate it as

$$E(X(t)|S=1) \approx \exp\{\mu(t)+\Sigma_V(t,t)/2+\Sigma_W(t,t)/2\}, \tag{30}$$

where $\Sigma(t,t') = \Sigma_V(t,t') + \Sigma_W(t,t')$.

Using this approximation, estimates of the two covariance functions are obtained through the conditional joint probabilities,

$$Pr\{X_{ij}(t)X_{ij'}(t')=1|S_{ij}S_{ij'}=1\} \quad \approx \quad b(t)b(t')\exp\{\Sigma_V(t,t')\}$$

where $j \neq j\prime$ and $b(t)$ is used to denote the quantity $\exp\{\mu(t)+\Sigma_V(t,t)/2+\Sigma_W(t,t)/2\}$, for notational purposes and out of brevity. Similarly, the joint conditional probability of individuals from the same time period $j = j'$ is approximated as

$$Pr\{X_{ij}(t)X_{ij}(t') = 1|S_{ij} = 1\} \approx b(t)b(t')\exp\{\Sigma_V(t,t') + \Sigma_W(t,t')\}.$$

Using these two approximations, estimates of the mean function $\mu(\cdot)$, and two covariance functions, $\Sigma_V(\cdot,\cdot)$ and $\Sigma_W(\cdot,\cdot)$, are obtained in following steps. First, we implement a working assumption of independence across i and j. Then, for all $t_p \neq t_{p'}$ denote by

$$\begin{aligned}
\widetilde{E}_W(t_p,t_{p\prime}) &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{J_i-1}\sum_{j'>j}\left(X_{ijp}-X_{ij'p}\right)\left(X_{ijp'}-X_{ij'p'}\right)\widetilde{S}_{ij}\widetilde{S}_{ij'}}{2\sum_{i=1}^{n}\sum_{j=1}^{J_i-1}\sum_{j'>j}\widetilde{S}_{ij}\widetilde{S}_{ij'}}, \\
\widetilde{E}_V(t_p,t_{p\prime}) &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{J_i-1}\sum_{j'>j}X_{ijp}X_{ij'p'}\widetilde{S}_{ij}\widetilde{S}_{ij'}}{\sum_{i=1}^{n}\sum_{j=1}^{J_i-1}\sum_{j'>j}\widetilde{S}_{ij}\widetilde{S}_{ij'}}
\end{aligned}$$

where $X_{ijp} = X_{ij}(t_p)$ and $\widetilde{S}_{ij} = \mathbb{1}_{\sum_{p=1}^{m}X_{ijp}>0}$. We estimate smooth versions of the estimators by applying a smoother onto the two sets of estimated values $\{(t_p,t_{p'}),\widetilde{E}_V(t_p,t_{p\prime}),p,p' = 1,\ldots,m \text{ and } p \neq p'\}$ and $\{(t_p,t_{p'}),\widetilde{E}_W(t_p,t_{p\prime}),p,p' = 1,\ldots,m \text{ and } p \neq p'\}$. We denote the smoothed estimators as $\widehat{E}_V(t,t\prime)$ and $\widehat{E}_W(t,t\prime)$. The number of eigenfunctions for the finite truncations, K_V and K_W, are selected by percentage of variation explained (PVE). In order to ensure semi-positive definite matrices, the negative eigenvalues are dropped in the spectral decomposition.

A smooth estimator of $b(\cdot)$, which as a reminder is the conditional expectation $E(X(t)|S = 1)$, is obtained by through evaluating for each observed t_p

$$\widetilde{b}(t_p) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{J_i}X_{ijp}\widetilde{S}_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{J_i}\widetilde{S}_{ij}}.$$

A smooth estimator of $b(\cdot)$ is obtained by applying a smoother onto estimated values $\{t_p : \widetilde{b}(t_p), p, p' = 1,\ldots,m\}$ and we denote this smooth estimator as $\widehat{b}(\cdot)$. Using these smooth estimates, we estimate the covariance functions $\Sigma_V(t,t')$ and $\Sigma_W(t,t')$ as

$$\widehat{\Sigma}_V(t,t') = \log\left(\frac{\widehat{E}_V(t,t')}{\widehat{b}(t)\widehat{b}(t')}\right) \tag{31}$$

$$\widehat{\Sigma}_W(t,t') = \log\left(\frac{\widehat{E}_W(t,t')}{\widehat{b}(t)\widehat{b}(t')}\right) - \widehat{\Sigma}_V(t,t'). \tag{32}$$

As outlined in 2.3, by decomposing these estimated covariance functions we can estimate the first and second level eigenfunctions and their corresponding eigenvalues,

$\left\{\widehat{\phi}_{V,k}, \widehat{\lambda}_{V,k}\right\}_{k\geq 1}$ and $\left\{\widehat{\phi}_{W,k}, \widehat{\lambda}_{W,k}\right\}_{k\geq 1}$. Finally the mean function is estimated by plugging in the estimators into equation 14 and solving for $\mu(t)$,

$$\widehat{\mu}(t) \quad = \quad \left[\log(\widehat{b}(t)) - \widehat{\Sigma}_V(t,t)/2 - \widehat{\Sigma}_W(t,t)/2\right].$$

We have now obtained the estimates for the mean function and sets of eigenfunctions through the exponential approximation. The Classification rules are developed following the procedures outlined in Section 2.3 and further described in Weishampel et al. (2021b), using these new estimated features: $\widehat{\mu}(\cdot)$, $\left\{\widehat{\phi}_{V,k}, \widehat{\lambda}_{V,k}\right\}_{k=1}^{K_V}$, and $\left\{\widehat{\phi}_{W,k}, \widehat{\lambda}_{W,k}\right\}_{k=1}^{K_W}$.