Statistics and Its Interface Volume 17 (2024) 127–142

A review of nonparametric regression methods for longitudinal data

Changxin Yang[*,†] and Zhongyi Zhu[†]

Longitudinal data, which involve measuring a group of subjects repeatedly over time, frequently arise in many clinical and biomedical applications. To identify the complex patterns of change in the outcome and their association with covariates over time, a sufficiently flexible model is always required. Nonparametric regression, known for being data-adaptive and less restrictive than parametric approaches, becomes a promising tool for handling longitudinal data. This paper reviews various nonparametric regression methods for longitudinal data, including specific traditional nonparametric methods for the univariate case and several representative methods for the multivariate case, among which tree-based techniques are dominant. We summarize their motivations and provide a brief practical performance comparison of these methods in simulations, as well as discuss potential future research directions.

Keywords and phrases: Longitudinal data, Repeated measurements, Nonparametric regression, Machine learning, Regression tree.

1. INTRODUCTION

Generally, a typical longitudinal data framework involves repeated measurements on N subjects, with the number of measurements varying for each subject and denoted by n_i for the ith subject. Each observation consists of measurements on time t, response variable y, and predictors $\mathbf{x}$. For simplicity, let $\mathbf{t}_i = (t_{i1}, \ldots, t_{in_i})^T$, $i = 1, ..., N$, where t_{ij} is the jth measurement time of the ith subject and analogously define $\mathbf{y}_i, \mathbf{x}_i$. Repeated measurements for each individual can provide crucial information about changes in outcome and covariates over time. This enables a more comprehensive understanding of the pattern of change over time and allows for better decision-making in various fields, such as medical treatment, economic modeling, weather forecasting, disaster warning, and more.

Analyzing longitudinal data presents significant challenges due to several unique features inherent in this type of data. One of the most troublesome features is that repeated measurements within each subject are typically related, and the average level of responses may vary between individuals. Ignoring these correlations and individual differences can result in significant estimation errors. In addition, the observation times for longitudinal data can be ordered in either equal or irregular spaces, which may limit the approaches that can be used for analysis. This may be caused by the data collection mechanism and missing values in repeated measurements. For a more in-depth discussion of the features of longitudinal data, refer to Liu [33].

To address these issues, statisticians have developed several parametric models, such as the linear mixed-effects model (LMM) introduced by Laird and Ware [26] and the marginal model for the generalized estimating equations (GEE) by Liang and Zeger [29]. Other acknowledged models include the generalized linear mixed model (GLMM) and the varying coefficient model (VCM). However, parametric models can suffer from misspecification when the data are highly nonlinear. Nonparametric regression, which is data-adaptive and less restrictive, has become a promising tool. Common non-parametric methods include kernel regression, spline methods, and local polynomial regression, among others. For more traditional methods for longitudinal data analysis, see Fitzmaurice et al. [13].

In recent years, machine learning methods, such as regression trees (Breiman [3]), boosting (Friedman [16]), neural networks have gained popularity due to their ability to identify complex relationships without making strong assumptions about the data distribution, especially in high-dimensional cases. Nevertheless, these methods often assume that observations are independently and identically distributed (*iid*), which is not the case for longitudinal data. In the past, it has been common practice to use repeated measures of the outcome variable as a vector response to generate multivariate trees. However, this approach typically requires data with a specific structure, such as balanced data, as demonstrated in Segal [45]. To resolve this dilemma, several machine learning methods have been combined with traditional longitudinal data models, such as the marginal model, linear mixed-effects model, and varying coefficient model. Examples of such studies include Pande et al. [37], Sela et al. [46], and Deshpande et al. [8].

In this article, we review several important nonparametric methods for analyzing longitudinal data and show how machine learning methods, specifically regression trees, can be applied to this type of data. In addition, we investigate which approach is recommended by simulations and

[*] Corresponding author.

[†] Department of Statistics and Data Science, Fudan University.

briefly discuss potential future research directions. As the structure of predictor variables may differ depending on the research questions, we consider both univariate and multivariate cases in this paper. Thus, we not only review methods that model the response variable y as a function of a scalar t, but also those that include other covariates. These covariates can be either time-invariant vector $\mathbf{x}_{ij} \equiv \mathbf{x}_{ij'}$ (such as gender and race) or time-varying covariate vector $\mathbf{x}_i = \{\mathbf{x}_{i1}, ..., \mathbf{x}_{in_i}\}$ (such as salary and temperature). We categorize different nonparametric methods according to the data structures in the following sections.

Our review is organized as follows. In Section 2, we review basic models for univariate longitudinal data and several nonparametric approximate methods. Section 3 focuses on existing nonparametric methods combined with regression trees for multivariate cases. After that, we provide simulation studies for various scenarios in Section 4. A concluding discussion and future research directions are given in Section 5.

2. SINGLE-COVARIATE NONPARAMETRIC REGRESSION

We begin by considering the traditional scenario where the predicted variable is univariate. In such cases, nonparametric regression usually only considers the potential relationship between the response variable y and the measurement time t. It is worth noting that t can be substituted for any scalar variable x. In this section, we will discuss several representative nonparametric estimation methods.

2.1 Basic models for single-covariate nonparametric regression

Nonparametric models for longitudinal data can be broadly categorized into two types. The first type inherits the principles of the LMM and employs nonparametric techniques to estimate both fixed and random effects in the traditional LMM model. The second type is the marginal model, which focuses more on fixed effects and relies on fewer assumptions for potential distributions.

A nonparametric mixed-effects (NPME) model for longitudinal data is introduced in Shi et al. [48]:

$$y_{ij} = f\left(t_{ij}\right) + v_i\left(t_{ij}\right) + \epsilon_{ij}, \tag{1}$$

where $f(t)$ models the population means, called the fixed-effects curve; $v_i(t)$ models individual curve variations from $f(t)$, called random-effects curves with expectation equal to 0 and a covariance function $\mathrm{E}\left[v_i(s)v_i(t)\right]$, ϵ_{ij} are measurement errors. It is expected that both the population and random effects curves can be approximated using some basis functions or methods. The NPME model has the advantage of incorporating both fixed and random effects, which allows for a more flexible and accurate representation of the underlying data structure and can better capture the heterogeneity in the data.

In some cases, we are more concerned with the prediction of y for a new individual whose random effect may be different from any individual in the sample. We would like to pay more attention to population level than individual-specific random effect, and a helpful remedy for it is the marginal model.

Specifically, suppose $\mathbf{y}_i$ marginally has the vector of conditional mean $\boldsymbol{\mu}_i = \left(\mu_{i1}, \ldots, \mu_{in_i}\right)^T$ and variance $\operatorname{var}\left(\mathbf{y}_i \mid \mathbf{t}_i\right) = \mathbf{\Sigma}_i$. A marginal nonparametric model is usually given by

$$\mu_{ij} = g(\theta\left(t_{ij}\right)), \quad \operatorname{var}\left(y_{ij} \mid t_{ij}\right) = \phi^{-1} v\left(\mu_{ij}\right), \tag{2}$$

where $\theta(\cdot)$ is an unknown smooth function and $g(\cdot)$ is a known monotonic link function, $v(\cdot)$ is a variance function and ϕ is a scale parameter.

Indeed, there are numerous well-known nonparametric approximate methods for nonparametric regression, such as local polynomial kernel estimates, local averaging estimates (including the kernel, partitioning, and nearest neighbor estimates), least squares estimates using splines, penalized least squares estimates, and so on. We will introduce some of them below, and for more information, refer to Wu and Zhang [62]. While the theoretical convergence of these methods has been established, it is beyond the scope of this paper. Therefore, we will not delve into further details on this topic.

2.2 Local polynomial kernel method

Based on (2), Lin and Carroll [30] proposed a kernel GEE (KGEE) estimator which extends the GEE method for parametric regression to the nonparametric case. Kernel GEE method approximates $\theta\left(t_{ij}\right)$ locally by a d th-order polynomial as $\theta\left(t_{ij}\right) \approx \beta_0 + \cdots + \beta_d\left(t_{ij} - t\right)^d = \mathbf{G}^T(t_{ij} - t)\boldsymbol{\beta}$, where $\mathbf{G}(z) = \left[1, z, \ldots, (z)^d\right]^T$, $\boldsymbol{\beta} = \left(\beta_0, \ldots, \beta_d\right)^T$. In what follows $f^{(1)}(t)$ denotes the first-order derivative of any arbitrary function $f(t)$ and analogously define $f^{(2)}(t)$. Let $\mathbf{G}_i(t) = \left\{\mathbf{G}(t_{i1} - t), \ldots, \mathbf{G}(t_{in_i} - t)\right\}^T$. The symmetric local polynomial kernel GEE estimating equation is written as

$$\sum_{i=1}^{N} \mathbf{G}_i(t)^T \mathbf{\Delta}_i(t) \mathbf{K}_{ih}^{1/2}(t) \mathbf{V}_i^{-1}(t) \mathbf{K}_{ih}^{1/2}(t) \left\{\mathbf{y}_i - \boldsymbol{\mu}_i(t)\right\} = 0, \tag{3}$$

where $\boldsymbol{\mu}_i(t) = \left\{\mu_{i1}(t), \ldots, \mu_{in_i}(t)\right\}^T$ with $\mu_{ij}(t) = g\left\{\mathbf{G}^T(t_{ij} - t)\boldsymbol{\beta}\right\}$, $\mathbf{\Delta}_i = \operatorname{diag}\left((g)^{(1)}\left[\mathbf{G}^T(t_{ij} - t)\boldsymbol{\beta}\right]\right)$, $\mathbf{K}_{ih}(t) = \operatorname{diag}\left\{K_h\left(t_{ij} - t\right)\right\}$ with a symmetric zero-mean kernel function $K_h(s) = h^{-1}K(s/h)$, $\mathbf{V}_i$ is an invertible working matrix, which can be estimated using the method of moments. The kernel functions can be chosen from a wide range of options such as the Gaussian, Epanechnikov, or triangular kernel. The bandwidth h determines the width of the kernel function and plays an important role in nonparametric regression. There are several approaches for selecting the bandwidth, including the least-squares cross-validation (CV) method, empirical bias bandwidth selection (EBBS) method in Ruppert [43], and so on.

After estimating $\boldsymbol{\beta}$ at time t, the estimated $\theta(t)$ is obtained as $\hat{\theta}(t) = \hat{\beta}_0$. Lin and Carroll proved that the kernel GEE estimator is most efficient when the working matrix $\mathbf{V}_i$ is an identity matrix $\mathbf{I}_i$ of dimension n_i. This contrasts with the parametric GEE method, indicating that the traditional kernel method cannot account for within-cluster correlation. Wang [54] provided an alternative kernel smoothing method based on the kernel GEE estimator, which is called the seemingly unrelated kernel (SUR) estimator, to better capture the correlation information.

Denoting kernel estimator at the wth iteration by $\widehat{\theta}_K^{[w]}(t)$, the updated estimator at the $(w+1)$th iteration is $\widehat{\theta}_K^{[w+1]}(t) = \widehat{\beta}_0$, while $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_0, \ldots, \widehat{\beta}_d)^T$ solves kernel-weighted estimating equation:

$$\begin{aligned} 0 = &\sum_{i=1}^{N} \sum_{j=1}^{n_i} K_h\left(t - t_{ij}\right) \\ &* \left[(g)^{(1)} \{\mathbf{G}^T(t_{ij}-t)\boldsymbol{\beta}\} \left(\mathbf{T}_{*j}^{i}\right)^{\mathrm{T}}\right] \\ &\times \left(\mathbf{V}_i\right)^{-1} \left[\mathbf{y}_i - \boldsymbol{\mu}_{*j}\left\{\mathbf{t}_i, \boldsymbol{\beta}, \widehat{\theta}_K^{[w]}(t)\right\}\right], \end{aligned}$$

where $\mathbf{T}_{*j}^{i}$ is a $n_i \times (d+1)$ matrix of zeros except that the jth row is $\left\{1, \left(t_{ij}-t\right), \ldots, \left(t_{ij}-t\right)^d\right\}$, and the lth element of $\boldsymbol{\mu}_{*j}\left\{\mathbf{t}_i, \boldsymbol{\beta}, \widehat{\theta}_K^{[w]}(t)\right\}$ is $g\left\{\mathbf{G}^T(t_{il}-t)\boldsymbol{\beta}\right\}$ when $l = j$, and is $g\left\{\widehat{\theta}_K^{[w]}(t_{il})\right\}$ when $l \neq j$.

When using the estimator of kernel GEE as the initial estimate $\widehat{\theta}_K^{[0]}(t)$ and $\widehat{\theta}_K^{[1]}(t)$ as a one-step update estimator, Wang [54] showed that the one-step update estimator, with great computational convenience, behaves almost as well as the fully iterated one. Additionally, the SUR estimator is very adaptable and can be generalized to a wide range of situations, such as handling multivariate $\mathbf{x}$ following Ruppert and Wand [44]. Furthermore, Wang et al. [55] and Lin et al. [31] incorporated the SUR method into semiparametric marginal models to achieve semi-parametric efficient estimation.

2.3 Local polynomial for mixed effect model

In the previous section, it is pointed out that kernel GEE could not fully utilize the information of the correlation matrix. Different from Wang [54], Wu and Zhang [61] proposed a method called the local polynomial linear mixed-effect (LLME) estimator to address this issue, which combines the local polynomial kernel method with the NPME model (1) to incorporate within-cluster correlation.

Similarly, it is assumed that $f(t)$ and $v_i(t)$ have $(d+1)$ th continuous derivatives. Thus for any fixed $t, f(t)$ and $v_i(t)$ at t_{ij} can be approximated by d th-order polynomials within a neighborhood of t as:

$$f\left(t_{ij}\right) \approx \beta_0 + \beta_1\left(t_{ij}-t\right) + \ldots + \beta_d\left(t_{ij}-t\right)^d = \mathbf{G}^T(t_{ij}-t)\boldsymbol{\beta},$$

and

$$v_i\left(t_{ij}\right) \approx b_{i0} + b_{i1}\left(t_{ij}-t\right) + \cdots + b_{id}\left(t_{ij}-t\right)^d = \mathbf{G}^T(t_{ij}-t)\mathbf{b}_i.$$

Then within a neighborhood of t, the model can be reasonably represented as an LMM model,

$$y_{ij} = \mathbf{G}^T\left(t_{ij}-t\right)\left(\boldsymbol{\beta} + \mathbf{b}_i\right) + \epsilon_{ij},$$

inferences for $\boldsymbol{\beta}$ and $\mathbf{b}_i$ follow the similar spirit of LMM. The only difference is that they take a local likelihood estimator with kernel weight instead of traditional maximum likelihood estimation. It has been shown that for bounded n_i and as $N \to \infty$, the asymptotic performance of the LLME estimator is similar to kernel GEE and it considers within-subject correlations more carefully.

Another way to estimate fixed effect curve and random effect curve is using B-spline basis functions, as described in Rice et al. [42]. See the next section for more details on the spline methods. The trade-off is computation time, as LLME is done locally at each point, while the B-spline approach is a global smoothing procedure.

2.4 Spline method

Besides the local polynomial kernel method, another popular nonparametric approach is the spline method, such as the smoothing spline (Green et al. [18]) and regression spline (Stone et al. [51]), which are defined as combinations of some basic functions such as B-spline basis. Lin et al. [32] showed that the smoothing spline estimator is asymptotically equivalent to the SUR kernel estimator and can be seen as a higher-order SUR kernel estimator. To illustrate the basic concepts of the spline methods, first rewrite the marginal model

$$E(y_{ij} \mid t_{ij}) = g(\theta\left(t_{ij}\right)) = F\left(t_{ij}\right), \tag{4}$$

where $F(t)$ is a smooth but unknown function. Denoted by $\{B_1(z), \ldots, B_q(z)\}$ a set of basis functions, one approximates $F(t)$ by $F(t) \approx \sum_{l=1}^{q} B_l(t)\alpha_l$, where q is determined by the number of knots and the order of the basic function. The selection of the order involves a trade-off between model complexity and prediction accuracy. Typically, the order of the spline function is selected through CV, and it commonly falls within the range of 2 to 4.

Given the desired order, the smoothing spline method estimates $F(t)$ using all observed values as knots, and thus, there is no need to select q. However, a value for the smoothing parameter λ needs to be optimized. The least squares estimators $\widehat{\alpha}_l$ of α_l are then obtained by minimizing a penalized sum of squares (PSS):

$$\sum_{i=1}^{N}\left[\mathbf{y}_i - F\left(\mathbf{t}_i\right)\right]^T \mathbf{V}_i^{-1}\left[\mathbf{y}_i - F\left(\mathbf{t}_i\right)\right] + \lambda \int_{t_{\min}}^{t_{\max}} \left[F^{(2)}(s)\right]^2 ds, \tag{5}$$

where $\lambda \geq 0$ is the smoothing parameter for smoothing spline, $F(\mathbf{t}_i) = (F(t_{i1}), \ldots, F(t_{in_i}))^T$ and $t_{\min}$ and $t_{\max}$ are the times of the first and last measurements across N subjects. The optimal value of λ depends on the characteristics of the data and can be selected by generalized cross-validation (GCV) and general maximum likelihood (GML).

When the sample size is large, the computational cost of smoothing splines can increase significantly. In this case, regression splines may be a more feasible alternative since they typically do not include a smoothing penalty and rely on fewer knots while selecting optimal knots is crucial. The number and position of knot points determine the potential flexibility of the regression spline. In practice, knots are usually placed at equally spaced intervals or at quantiles of the data. Criteria such as AIC and GCV can be used to select the number of knots, and a more comprehensive investigation of knot selection has been surveyed by Wand [53]. Zhu et al. [68] confirmed that the asymptotic bias for regression splines is robust to working matrix misspecification, whereas smoothing splines are not. Additionally, penalized splines (P-splines) combine the properties of regression splines and smoothing splines, reducing the computational burden of smoothing splines and being less sensitive to knot allocation. For more details, refer to Eilers et al. [9].

2.5 Functional principal components analysis approach

Longitudinal data, when n_i is large, is also known as functional data in engineering and biological applications. Functional principal component analysis (FPCA) is a commonly used method in functional data analysis, as discussed in Rice and Silverman [41]. In FPCA, the response variables $y_i(t)$ are treated as realizations of a smooth L^2 process with mean $\mu(t)$ and covariance function $\operatorname{cov}(y_i(s), y_i(t))$. The FPCA model can be expressed as follows:

$$y_{ij} = \mu\left(t_{ij}\right) + \sum_{l=1}^{\infty} \xi_{il} \phi_l\left(t_{ij}\right) + \epsilon_{ij}, \tag{6}$$

where $E(\epsilon_{ij}) = 0, \operatorname{var}\left(\epsilon_{ij}\right) = \sigma^2$, ϕ_l is the eigenfunctions for orthogonal expansion of covariance function $\operatorname{cov}(y_i(s), y_i(t))$ with the lth largest eigenvalue and ξ_{il} is principal component scores for the ith subject and lth eigenfunction. Briefly speaking, FPCA replaces the prespecified basis spline functions in Section 2.3 with a mean function and the eigenfunctions of the covariance operator of the response.

Under certain conditions, it has been proven that infinite-dimensional processes can be well approximated by the projection on the function space spanned by the first $\mathcal{K}$ eigenfunctions as $\hat{y}_{ij} = \hat{\mu}(t_{ij}) + \sum_{l=1}^{\mathcal{K}} \hat{\xi}_{il} \hat{\phi}_l(t_{ij})$ in Boente et al. [2]. Thus FPCA can effectively reduce the number of basis functions. Yao et al. [63] demonstrated that this method can be extended to situations in longitudinal data analysis and provide methods to estimate the eigenfunctions and principal components. To select the number $\mathcal{K}$ of the eigenbasis, one can choose the leave-one-subject-out CV method or AIC.

3. MULTIPLE-COVARIATE NONPARAMETRIC REGRESSION

In the previous sections, we presented several univariate regression methods for longitudinal data. However, when there are other covariates $\mathbf{x} \in R^p$ related to the response, one needs to take the influence of these possibly important covariates into account to avoid unnecessary loss of information. In this section, we extend our review from scalar t to the multivariate case. The traditional spline methods are not suitable for multidimensional problems, and the general kernel function will face the curse of dimensionality when the dimension of $\mathbf{x}$ increases. Therefore, follow-up research usually combines traditional models with machine learning methods such as regression trees. Additionally, during the repeated measurements, some covariates are measured only at baseline, referred to as time-invariant covariates $\mathbf{x}_{ij} \equiv \mathbf{x}_{ij'}$ while others are measured along with the response referred to as time-varying covariates. To avoid notation confusion, we denote time-invariant covariates as $\mathbf{x}_i^* \in R^p$ to distinguish from $\mathbf{x}_i$. Methods for analyzing multivariate longitudinal data may vary depending on the structure of the covariates.

Importantly, it should be noted that although the practical effectiveness of random forests has been extensively validated in the analysis of longitudinal data, the theoretical analysis of such composite methods seems rather difficult and remains an open issue.

3.1 Regression trees combined with marginal model

The marginal models approach to analyzing multivariate longitudinal data aims to estimate the mean function, while treating the correlations between repeated measurements of the same individual as noise correlations. Based on such models, regression trees are usually constructed by directly considering the correlations within the same individual. A typical approach involves finding the best split by minimizing the (weighted) sum of squared residuals when splitting a node.

3.1.1 Splinetree

Segal [45] firstly proposed a kind of longitudinal regression tree in which the repeated measures for the response variable $\mathbf{y}_i = \{y_{i1}, ..., y_{im}\}$ are assumed to have the same number of equally spaced measurements and are treated as multiple responses to build multivariate trees with modified split criteria. However, naively applying multivariate decision trees to long vector responses is not successful in some cases and the assumption of balanced data seems too restrictive in real life. Yu and Lambert [64] extended this work by reducing the dimension of the outcome vector, which is called splinetree.

Formally, assuming the data structure is $\{(y_{ij}, t_{ij}, \mathbf{x}_i^*)\}$, splinetree mainly explores the relationship between the time-invariant covariate and the response variable. Instead of directly taking repeated measures of each individual response variable as a response vector to build a multivariate tree, each individual's response curve is firstly represented as a linear combination of spline basis in splinetree:

$$\mu_{ij}(t_{ij}, \mathbf{x}_i^*) = \textstyle\sum_{l=1}^{q} \beta_{il}(\mathbf{x}_i^*) B_l(t_{ij}), \tag{7}$$

and the coefficient vectors $\{\boldsymbol{\beta}_i(\mathbf{x}_i^*) = (\beta_{i1}(\mathbf{x}_i^*), \ldots, \beta_{iq}(\mathbf{x}_i^*))^T\}_1^N$ are estimated by minimizing PSS in (5) with $\mathbf{V}_i = \mathbf{I}_i$. The selection of q and λ is similar to the smoothing spline in Section 2.4. The estimated coefficients $\{\boldsymbol{\beta}_i(\mathbf{x}_i^*)\}_1^N$ are treated as new response variables to build trees and the prediction error in the node $\mathcal{L}$ is measured by the standardized squared error loss:

$$SS_{\mathcal{L}} = \sum_{i \in \mathcal{L}} \left(\boldsymbol{\beta}_i - \overline{\boldsymbol{\beta}}_{\mathcal{L}}\right)^T \mathbf{B}^T \mathbf{B} \left(\boldsymbol{\beta}_i - \overline{\boldsymbol{\beta}}_{\mathcal{L}}\right),$$

where $\mathbf{B}$ is the $M \times q$ basis matrix evaluated at the chosen fixed time points $\{t_1, \ldots t_M\}$, $\overline{\boldsymbol{\beta}}_{\mathcal{L}}$ represents the mean coefficient vector of all individuals assigned to node $\mathcal{L}$, which is used to determine the predicted curve at this node. The splitting rule is to divide each individual into different nodes to minimize prediction error as much as possible.

This method combines the computational efficiency and model-free characteristics of trees with the advantages of smoothing splines. It assumes that different individuals are independent, making it reasonable to fit a multivariate tree with an individual-specific growth curve. However, the splinetree method does not take into account the influence of the covariance matrix when estimating $\boldsymbol{\beta}$ and growing trees, which may decrease estimation efficiency. Additionally, Neufeld [35] used regression splines instead of smoothing splines to reduce the computational burden. Other works, such as Lee [27, 28], had proposed similar multivariate decision tree methods that use GEE techniques within each node for general types of response variables. Combining some of the methods mentioned in Section 2 with these multivariate trees could be an interesting direction for future research.

3.1.2 Boostmtree

In addition to the tree-based method, boosting, another machine learning approach, has attracted attention in recent years with its application in various data types. Pande et al. [37] put forward a boosted multivariate tree (called boostmtree) for longitudinal data which assumes a similar data structure to splinetree.

Specifically, it is assumed that the vector of conditional mean satisfies

$$\boldsymbol{\mu}_i(\mathbf{t}_i, \mathbf{x}_i^*) = \beta_0\left(\mathbf{x}_i^*\right) \mathbf{1}_i + \sum_{l=1}^{q} \mathbf{B}_l\left(\mathbf{t}_i\right) \beta_l\left(\mathbf{x}_i^*\right) = \mathbf{F}_i(\boldsymbol{\beta}(\mathbf{x}_i^*)), \tag{8}$$

where $\mathbf{1}_i = (1, \ldots, 1)^T_{n_i \times 1}$, $\{\mathbf{B}_l(\mathbf{t}_i)\}_1^q$ are the basic function vectors evaluated at $\mathbf{t}_i$, and $\boldsymbol{\beta}(\mathbf{x}_i^*) = (\beta_0(\mathbf{x}_i^*), \ldots, \beta_q(\mathbf{x}_i^*))^T$. For boostmtree, the common choices of basic function are cubic B-spline and equally spaced knots. The optimal number of knots is selected on a grid of intervals.

Following the framework in Friedman [16], boostmtree starts with an initial value $\boldsymbol{\beta}^{(0)}(\mathbf{x}^*)$, and the value at iteration $w = 1, \ldots, W$ is updated from the previous value according to

$$\boldsymbol{\beta}^{(w)}(\mathbf{x}^*) = \boldsymbol{\beta}^{(w-1)}(\mathbf{x}^*) + v\mathbf{h}\left(\mathbf{x}^*; \mathbf{a}_w\right),$$

$$\boldsymbol{\mu}_i^{(w)} = \mathbf{F}_i\left(\boldsymbol{\beta}^{(w)}(\mathbf{x}^*)\right),$$

where $0 < v \leq 1$ is a learning parameter, $\mathbf{h}\left(\mathbf{x}^*; \mathbf{a}_w\right)$ denotes an optimized base learner over $\mathbf{a} \in A$ and A is the set of parameters of the weak learner. For a regression tree, A refers to the splitting variables, split locations (which actually represent the information of the generated nodes), and terminal node predictors. The goal of boosting is to update $\boldsymbol{\beta}(\mathbf{x}^*)$ at the wth iteration by minimizing:

$$\sum_{i=1}^{N} L_i\left(\mathbf{y}_i, \boldsymbol{\mu}_i^{(w)}\right),$$

where $L_i\left(\mathbf{y}_i, \boldsymbol{\mu}_i\right) = \left(\mathbf{y}_i - \boldsymbol{\mu}_i\right)^T \mathbf{V}_i^{-1}\left(\mathbf{y}_i - \boldsymbol{\mu}_i\right), \mathbf{V}_i$ is the working matrix estimated by an in-sample CV method in boostmtree for each iteration. The most important difference between boostmtree and the traditional boosting method is that it takes into account the covariance matrix when boosting $\boldsymbol{\beta}(\mathbf{x}^*)$.

To solve the minimization problem, a simple idea is to find the optimal descent gradient for subject i with respect to $\boldsymbol{\beta}\left(\mathbf{x}_i^*\right)$ evaluated at $\boldsymbol{\beta}^{(w-1)}\left(\mathbf{x}_i^*\right)$ as

$$\mathbf{g}_{w,i} = -\left.\frac{\partial L_i\left(\mathbf{y}_i, \boldsymbol{\mu}_i\right)}{\partial \boldsymbol{\beta}\left(\mathbf{x}_i^*\right)}\right|_{\boldsymbol{\beta}\left(\mathbf{x}_i^*\right) = \boldsymbol{\beta}^{(w-1)}\left(\mathbf{x}_i^*\right)}.$$

However, this gradient is defined only at training data $\{\mathbf{x}_i^*\}_1^N$ and cannot be generalized to other $\mathbf{x}^*$-values. Boostmtree fits a multivariate regression tree to determine the ℓ_2-closest base learner to the gradient at any $\mathbf{x}^*$ with a two-stage procedure.

In the first step, a multivariate regression tree $\mathbf{h}(\mathbf{x}^*; a)$ is fitted based on Ishwaran et al. [24], using $\{(\mathbf{g}_{w,i}, \mathbf{x}_i^*)\}_1^N$ as training data, a including $\{\mathcal{L}_{k,w}\}_1^K$, the total K terminal nodes of the regression tree at the wth iteration. In the second step, while keeping all other parameters in a fixed, the terminal node predictors are optimized as

$$\{\boldsymbol{a}_w\} = \underset{a \in A}{\operatorname{argmin}} \sum_{i=1}^{N} L_i\left(\mathbf{y}_i, F_i\left(\boldsymbol{\mu}_i^{(w-1)} + \mathbf{h}\left(\mathbf{x}_i^*; \boldsymbol{a}\right)\right)\right),$$

where $\mathbf{h}(\mathbf{x}^*; \boldsymbol{a}) = \sum_{k=1}^{K} \boldsymbol{\gamma}_{\mathbf{k}} 1(\mathbf{x}^* \in \mathcal{L}_{k,w})$, $\boldsymbol{\gamma}_k$ is the terminal node predictor of $\mathcal{L}_k$. In addition, boostmtree can add a smooth penalty at the second step to enhance its generalization ability.

Through simulation, Pande et al. [37] showed that this method is competitive when the underlying model contains high-order interactions, and it is robust to covariance misspecification. However, an obvious flaw of splinetree and boostmtree is that all observations of the same individual need to be assigned to the same node to describe the growth curve when dividing. Therefore, it can only be applied to features that do not change over time. As there is no satisfactory result to provide a tree-splitting rule to maintain all observations for each subject after splitting, Pande et al. [36] proposed a different boosting base learner with B-spline to handle time-varying covariates for longitudinal data. In addition to marginal models, there are more boosting methods for longitudinal data in Yue et al. [65] based on the varying-coefficient model, and Sigrist [49, 50] based on mixed effects models.

3.1.3 Historical tree

A natural approach to modifying a regression tree for longitudinal data is to make full use of historical information when splitting nodes. Sexton et al. [47] provided such a vehicle called historical tree as an attempt to estimate how the response depends on its prior realizations as well as time-varying predictor variables.

More specifically, the training data is denoted as $\{y_{ij}, t_{ij}, \mathbf{x}_{ij}\}$, with time-invariant predictors classified into the concurrent group and time-varying predictors in both the concurrent and historical groups. In a historical regression tree, splitting on a concurrent predictor follows the CART criterion used in Breiman [3]. For historical predictors, a summary function is used to capture historical information before the splitting. One example of a summary function is

$$s\left(f, \{\mathbf{z}_{ij}\}, k\right) = \sum_{t_{il} \in [t_{ij} - f_1, t_{ij})} I\left(z_{ilk} < f_2\right), \quad l = 1, \ldots, j,$$

where $\{\mathbf{z}_{ij}\} = \{\mathbf{z}_{il} = (y_{il}, \mathbf{x}_{il}) : t_{il} < t_{ij}\} \in R^{p+1}$ denotes historical values of subject i prior to time t_{ij}, and z_{ilk} is its kth component, $f = (f_1, f_2)$ is the argument vector of the summary function. Given a node $\mathcal{L}$, the splitting based on a historical predictor is done by solving

$$\underset{(k,\mu_L,\mu_R,c,f)}{\operatorname{argmin}} \sum_{(ij)\in\mathcal{L}} \{((y_{ij} - \mu_L I\left(s\left(f, \{\mathbf{z}_{ij}\}, k\right) \leq c\right) - \mu_R I\left(s\left(f, \{\mathbf{z}_{ij}\}, k\right) > c\right)\}^2.$$

Each node of the historical tree searches for the best split among all splits of concurrent and historical predictors.

One explanation for historical tree is that samples with close measurement times may have a higher correlation, and incorporating their values can improve the performance of the trees. However, determining the optimal cut-off point for time-varying predictors can be computationally expensive as there are additional parameters to be optimized. Additionally, there is a lack of historical information when predicting for a new individual, which can affect prediction performance. Besides, due to the limitations of the summary function, historical information may not be fully utilized. More comprehensive methods are needed to fully leverage the rich information in longitudinal data.

3.2 Regression trees combined with LMM

The characteristic of the LMM model is that y is modeled by fixed and random effects, and the correlation within the individual is explained by the random effect. Thus, one classic idea is to use regression trees to estimate the fixed effect in the LMM model: once we know the random effect, the fixed effect can be estimated by regression trees as *iid* data. Besides, Rabinowicz et al. [40] recently gave a new way to account for the correlation directly when growing LMM-based trees.

3.2.1 Mixed effects regression tree

Rather than methods based on marginal models like splinetree and boostmtree, a popular trend in recent years is to combine LMM with regression trees which allow observations within clusters to be split and handle time-varying covariates as:

$$y_{ij} = f\left(\mathbf{x}_{ij}\right) + \mathbf{Z}_{ij}^T \mathbf{b}_i + \epsilon_{ij}, \tag{9}$$

where $f(\mathbf{x}_{ij})$ refers to the fixed effect in the LMM estimated by regression tree, $\mathbf{Z}_{ij} \in R^r$ is the random effects covariate, $\mathbf{b}_i \sim N_r(0, \mathbf{D})$ is the random effects vector, and $\boldsymbol{\epsilon}_i = \{\epsilon_{i1}, ..., \epsilon_{in_i}\}^T \sim N(0, \sigma^2 I_{n_i})$ is the noise. The essential idea is to remove the random effect from y properly, and the remaining fixed effects with noise are irrelevant, then are used as new responses to train the regression trees as *iid* data. Since neither the random effects nor the fixed effects are known in advance, the estimation is alternatively updated, which is similar to the EM algorithm:

- Step 0: Initialize $\widehat{\mathbf{b}}_i = 0, \widehat{\mathbf{D}} = \mathbf{I}_r, \widehat{\sigma} = 1$;
- Step 1: Denote $\tilde{y}_{ij} = y_{ij} - \mathbf{Z}_{ij}^T \widehat{\mathbf{b}}_i$, train a regression tree with samples $\{\tilde{y}_{ij}, x_{ij}\}$, denote the tree predictions as $\hat{f}(x)$ for any point x;
- Step 2: Given $\hat{f}(x_{ij})$ predicted by Step 1, fit the linear model $\mathbf{y}_i = \hat{f}\left(\mathbf{x}_i\right) + \mathbf{Z}_i^T \mathbf{b}_i + \boldsymbol{\epsilon}_i$, update $\widehat{\mathbf{b}}_i, \widehat{\mathbf{D}}$, and $\widehat{\sigma}$ with the corresponding maximum likelihood estimation;
- Step 3: Repeat Step 1 and Step 2 until convergence, which is monitored by computing the generalized log-likelihood (GLL) criterion at each iteration:

$$GLL(f(\cdot), \mathbf{b}_i \mid y) = \textstyle\sum_{i=1}^N \{\mathbf{e}_i^T \mathbf{e}_i + \mathbf{b}_i^T \mathbf{D}^{-1} \mathbf{b}_i + \log|\mathbf{D}|\},$$

where $\mathbf{e}_i = \mathbf{y}_i - f\left(\mathbf{x}_i\right) - \mathbf{Z}_i^T \mathbf{b}_i$.

Such a method is called the Mixed effects regression tree (MERT) in Hajjem et al. [19]. When the random effects part can be effectively estimated, such algorithms perform well. Similar work has also been done in the Random effects/EM (REEM) tree introduced by Sela et al. [46] and the difference is that the REEM tree replaces the tree prediction at each terminal node $\mathcal{L}_k$ with local fixed effects μ_k by fitting the linear mixed effects model:

$$y_{ij} = \mathbf{Z}_{ij}^T \mathbf{b}_i + \sum_{k=1}^{K} I\left(\mathbf{x}_{ij} \in \mathcal{L}_k\right) \mu_k + \varepsilon_{ij}, \tag{10}$$

where K is the total number of terminal nodes in the tree. Capitaine et al. [6] extended the REEM tree to the forest and showed that both MERT and REEM tree and forest are applicable to the high-dimensional cases. Compared with the marginal model, this type of method can better predict future observations for those already included in the sample because their random effect has been estimated.

3.2.2 Generalized linear mixed-effects model tree

Instead of using piecewise constants to estimate the fixed effects in the REEM tree, Fokkema et al. [14] studied a generalized linear mixed-effects model tree (GLMM tree) which allows differences not only in intercepts across terminal nodes but also in slopes associated with $\mathbf{x}$. Firstly select some of the features of $\mathbf{x} \in R^p$ as the regression variable $\mathbf{s}^{rg}$, and some as the splitting variable $\mathbf{s}^{sp}$. The overlap between the two sets of variables is allowed. The GLMM predictor can be presented as:

$$g(y_{ij}) = \sum_{k=1}^{K} I\left(\mathbf{s}_{ij}^{sp} \in \mathcal{L}_k\right) \boldsymbol{\beta}_k^T \mathbf{s}_{ij}^{rg} + \mathbf{Z}_{ij}^T \mathbf{b}_i + \epsilon_{ij}. \tag{11}$$

Compared with the REEM tree, this approach has made a trade-off between the traditional random forest piecewise constant prediction and the LMM model and retains some interpretability.

The process of estimating parameters with the GLMM tree is similar to that of the REEM tree, both of which alternate between updating random effects and fixed effects. However, when growing trees to estimate the fixed effect, the GLMM tree replaces the CART criterion with model-based recursive partitioning (MOB) in Zeileis et al. [66] as splitting rules, which cycles iteratively through several steps:

- (1) With the current random effect estimator known, fit a GLMM to the dataset within the parent node to estimate $\boldsymbol{\beta}$,
- (2) make a test for parameter instability with respect to each of a set of partitioning variables and choose the variable associated with the highest instability,
- (3) find the best cut-point by minimizing the sum of the loss functions in child nodes,
- (4) repeat the procedure in each of the resulting subgroups until a stopping criterion is reached.

This tree-based method relies more on the specification of predictor variables than REEM, which may increase the danger of model misspecification. Nevertheless, it improves the prediction accuracy when the fixed-effects predictor variables are correctly specified or when the sample size is sufficient. A work similar to the GLMM tree is Bürgin et al. [4], where they used the maximum likelihood equation as the criterion for tree partitioning.

3.2.3 Tree-based LMM

Both the REEM tree and GLMM tree use a two-step estimation process, where regression trees are trained with *iid* data by removing the random effect for the response. However, Rabinowicz et al. [40] had developed a new regression tree for correlated data, called RETCO, which explicitly takes the correlation structure into account in the splitting criterion. It is well known that the CV with squared error loss is biased in cases involving non-*iid* data. To address this, Rabinowicz et al. [39] introduced a bias-corrected CV estimator that adds a correction term to the L_2 loss function. Rabinowicz et al. [40] applied this idea to the best splitting measurements for regression trees.

In particular, given the current nodes set $\mathcal{L}^{K^*} = \{\mathcal{L}_k\}_1^{K^*}$, RETCO finds the best node $(\tilde{k})$, the best covariate $\left(l_{\tilde{k}}\right)$ and the best threshold $\left(c_{\tilde{k}}\right)$ for splitting as follows:

$$\begin{aligned}\tilde{k}, l_{\tilde{k}}, c_{\tilde{k}} = \underset{k \in \mathcal{L}^{K^*}, l \in J_k, c \in \mathbb{R}}{\operatorname{argmin}} & \{\|\mathbf{y} - \widehat{\mathbf{y}}\|_2^2 \\ & + 2\operatorname{tr}\left(\mathbf{H}\left(\operatorname{Var}(\mathbf{y}) - \operatorname{Cov}\left(\mathbf{y}^*, \mathbf{y}\right)\right)\right)\},\end{aligned}$$

where $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}^T$, $\mathbf{y}^*$ is the new observations measured at the same covariate values as $\mathbf{y}$, $\operatorname{Cov}\left(\mathbf{y}^*, \mathbf{y}\right)$ is specified according to the research questions, J_k is the set of available covariates for splitting, and $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$ is the generalized least squares estimator of (10) given the nodes $\mathcal{L}^{K^*}/\mathcal{L}_k, \{\mathcal{L}_k \cap x_l \leq c\}$ and $\{\mathcal{L}_k \cap x_l > c\}$. It is noticed that the optimal split is searched through all terminal nodes at each step rather than splitting each node independently. The reason is that splitting one node may affect the splitting of others due to the correlation between observations from different nodes. Meanwhile, RETCO introduces a new stopping criterion whereby the loss function after splitting should not be higher than before splitting, with the aim of reducing computational burden.

RETCO directly takes the correlation structure into account in the splitting rules, estimating process, and stopping criterion to improve its prediction performance. Its loss function can become a new tool to develop more tree-based methods for correlated data. However, RETCO iterates through all nodes at each split, resulting in a significant increase in computational cost. Its additional stopping criteria result in fewer generated nodes, making it unsuitable for estimating continuous fixed effects and often only identifying strong signals.

3.3 VCM and its combination with tree

In addition to the marginal model and LMM for longitudinal data, there also exists a traditional model which has a meaningful interpretation and retains flexibility. This model is referred to as a varying coefficient model:

$$y(t) = \mathbf{x}^T(t)\boldsymbol{\beta}(t) + \epsilon(t).$$

In such a model, each component of the coefficient $\beta_j(t)$ is a function of the conditional variable t and describes the relationship between the response and covariate over time. Compared with LMM, it allows the constant parameters to evolve with certain characteristics and capture the dynamical pattern of this relationship.

Similar to the methods we reviewed in Section 2, the common practice to approximate $\boldsymbol{\beta}(t)$ is to use a linear combination of basic functions. One widely acknowledged approach is the smoothing spline method proposed by Hoover et al. [20] to minimize the penalized least squares criterion:

$$\sum_{i=1}^{N}\sum_{j=1}^{n_i}\left[y_{ij} - \left\{\sum_{l=1}^{p} x_{ijl}\beta_l\left(t_{ij}\right)\right\}\right]^2 + \sum_{l=1}^{p}\lambda_l \int \left\{\beta_l^{(2)}(t)\right\}^2 dt,$$

where $\left(\lambda_1, \ldots, \lambda_p\right)^{\mathrm{T}}$ are smoothing parameters . For each $l, \beta_l\left(t\right) = \sum_{k=1}^{q}\gamma_{lk}B_k(t)$, and x_{ijl} is the lth component of $\mathbf{x}_{ij}$. Parameter optimization is the same as in Section 2.4. A follow-up of their work is the polynomial spline proposed by Huang et al. [23] and the asymptotic distributions are derived. In addition to these two types of methods, the third popular type is the kernel approach which can be found in Wu et al. [59], Fan et al. [11]. Lately, Wang et al. [57] proposed an adaptive spline fitting method and showed that the new method can achieve significantly smaller mean squared errors.

A somewhat puzzling phenomenon observed is that many varying-coefficient-based methods ignore the correlation structure when estimating $\boldsymbol{\beta}(t)$, even if some of them are actually designed for longitudinal data, such as Huang et al. [23], Hoover et al. [20], and Wu et al. [60], as well as kernel approach. The kernel approach might be similar to kernel GEE, which cannot incorporate the within-subject correlation. Another reason may be the lack of estimation methods for the correlation structure in the early stages, and a misspecified working matrix could increase the mean squared error. Fan et al. [10] and Sun et al. [52] provided a systematic study of the estimation of the within-subject correlation structure with semiparametric varying-coefficient models. For a more detailed review of major methodological and theoretical developments on varying coefficient models, see Fan and Zhang [12].

There have been several studies on combining regression trees with VCM for *iid* data. Wang et al. [56] and Zhou et al. [67] developed boosted trees to estimate the varying coefficient $\boldsymbol{\beta}(\mathbf{s})$, where $\mathbf{s}$ is a multidimensional variable that is not limited to scalar time t. The latter work also proved theoretical consistency under mild assumptions. More recently, Deshpande et al. [8] proposed a method called VC-BART, which uses Bayesian trees to approximate $\boldsymbol{\beta}(\mathbf{s})$ and can handle correlated observations.

3.4 Other nonparametric regression approaches

So far, we have briefly discussed several nonparametric approaches for longitudinal data, mainly related to regression trees. These methods mainly focus on improving the accuracy of prediction. However, there are other works that focus on different aspects rather than prediction performance. For example, Calhoun et al. [5] proposed repeated measures random forests (RMRF), which focus on finding informative variables associated with y. This type of work is essential for medical and economic research. At each node, a random variable and a random cut-point are selected, and the candidate split is accepted or rejected based on hypothesis testing about the significance of parameters in GEE after splitting. Simulation results show that RMRF captures informative variables more often than naive regression trees.

Furthermore, the REEM tree and GLMM tree treat fixed effects as new responses for training. However, this idea need not be limited to regression trees. For example, Mandel et al. [34] offered a generalized neural network mixed model (GNMM) that replaces the linear fixed effect with the output of a feed-forward neural network and provided a bounds analysis for prediction error. Therefore, this direction deserves further study in the future. In addition to our review, for more details on tree-based methods for longitudinal data, see Hu et al. [22].

4. SIMULATION

The goal of this section is to compare the practical performance of the methods described above. As discussed in Sections $2-3$, each method may have different assumptions for the underlying model, and the estimation method may also vary accordingly, leading to differences in estimation efficiency under different scenarios. We investigate the estimation efficiency under the L_1 loss of these methods, as the L_2 loss has been extensively studied in previous literature. Generally speaking, L_2 is much more sensitive to outliers, while L_1 is less sensitive and hence more stable. We define the mean absolute error (MAE) as follows:

$$\mathrm{MAE} = \sum_{i=1}^{n_{test}} |y_i - \hat{y}(x_i)|/n_{test}.$$

Specifically, we introduce our simulation settings for univariate and multivariate cases in Sections 4.1 and 4.2, and present the numerical comparisons in Section 4.3. It is challenging to conduct comprehensive theoretical analyses of the efficiency of these methods, and we will not discuss them in this article. Thus, the numerical simulation results can briefly reflect these properties.

For each configuration, we generated 200 random datasets from the model:

$$y_{ij} = \mu\left(x_{ij}\right) + \epsilon_{ij}.$$

4.1 Simulation models of univariate case

Because one of our goals is to compare the performance of each method under different covariance specifications, we assume that the covariance matrix is known and optimize the smoothing parameter λ for the smoothing spline, the bandwidth parameter h for the kernel function, and the number of knots in regression splines using the same criterion, by minimizing the MAE on a grid of values. When calculating the kernel estimates, we use the Epanechnikov kernel function, which is defined as $K\left(x_{ij} - x\right) = \left(1 - |x_{ij} - x|^2\right)_+$. We consider the case where $n_i = m$ and investigate how the efficiency of different methods changes with the sample size N, m, and correlation coefficient ρ.

The noise, $(\epsilon_{i1}, \ldots, \epsilon_{im})$, is generated from $N(0,1)$ and one of the four correlation structures is used as follows:

- EX: exchangeable with a common correlation ρ;
- AR: first-order autoregressive process with correlation ρ;
- US: consider this matrix only when $m = 3$, unstructured with $\rho_{12} = \rho_{23} = \rho$ and $\rho_{13} = 0.5$, where ρ_{jk} is the correlation between ϵ_{ij} and $\epsilon_{ik}, j \neq k$;
- ID: zero correlations, i.e. independence.

In recent years, it has been found that the random forests estimator can be viewed as a weighted average of the training responses, and the weight is defined as:

$$w_{\mathbf{x}}\left(\mathbf{x}_i\right) = \frac{1}{B}\sum_{k=1}^{B} \frac{\mathcal{I}\left(\mathbf{x}_i \in \mathcal{L}(k, \mathbf{x})\right)}{|\mathcal{L}(k, \mathbf{x})|}, \tag{12}$$

where B is the total number of regression trees and $\mathcal{L}(k, \mathbf{x})$ is the leave containing $\mathbf{x}$ of the kth tree. Random forests are a promising tool for high-dimensional statistical learning, and a natural idea would be to replace the traditional kernel function with random forests weight which can avoid the curse of dimensionality caused by the kernel function. Thus we also explore what will happen if we replace the kernel function in the SUR estimator with the distributional random forest weight introduced in Cevid et al. [7] which is more concerned about the homogeneity of the distribution within the node when building forests. We denote it as the SUR-RFW method.

Scenario 1. We evaluate the performance of each method under various types of smooth models as well as a non-smooth model. The predictors x_{ij} are generated from $\mathrm{Un}(-2, 2)$. We select the following nonlinear functions:

- Model 1:
  $\mu(z) = \{z(1-z)\}^{1/2} \sin\left\{2\pi\left(1 + 2^{-3/5}\right) / \left(z + 2^{-3/5}\right)\right\}$,
- Model 2:
  $\mu(z) = \{z(1-z)\}^{1/2} \sin\left\{2\pi\left(1 + 2^{-7/5}\right) / \left(z + 2^{-7/5}\right)\right\}$,
- Model 3:
  $\mu(z) = \sin(8z - 4) + 2\exp\left\{-256(z - 0\cdot 5)^2\right\}$,
- Model 4:
  $\mu(z) = 0.5I(z \leq 0.5) - 0.5I(z > 0.5)$,

where $z = (x + 2)/4$.

We present the MAE for different true working matrices, which include AR, EX, US, and ID, as well as the MAE ratios for using a true working matrix (AR, EX, US) compared to the independent one. This investigation allows us to assess the average error of each method for different settings and the influence of using the real working matrix.

4.2 Simulation models of multivariate case

In this section, we introduce additional variables and compare the performance of each method using baseline covariates in Scenario 2 and time-varying covariates in Scenario 3. We evaluate how changes in sample size, dimensionality, and smoothness of the underlying models impact the methods' performance. For each model, let $\epsilon_{ij} = v_i + e_{ij}$, $e_{ij} \overset{i.i.d.}{\sim} N(0,4), v_i \sim N(0,4)$ and use a standardized L_1 loss in Scenario 2 and 3:

$$\mathrm{SMAE} = \frac{\mathrm{MAE}}{\hat{\sigma}(y)},$$

where $\hat{\sigma}(y)$ is the empirical standard deviation of the response.

Scenario 2. We assess the performance of different methods using both the smooth model from Pande et al. [37] and the non-smooth model from Pande et al. [36] for baseline covariates. Although it may appear similar to Scenario 1, where the change in $\mathbf{y}_i$ is solely dependent on t for each individual, given $\mathbf{x}_i^*$, in this case, the $\mathbf{x}_i^*$ values differ for each individual. Consequently, the marginal distribution varies from one individual to another.

Model 1 for baseline covariates:

$$\begin{aligned}\mu_{ij}(\mathbf{x}_i^*, t_{ij}) = 1.5 + 2.5x_{i1} + 10x_{i3} - 0.2\exp\left(x_{i4}\right) \\ - 0.65t_{ij}^2\left(x_{i2}\right)^2 x_{i3},\end{aligned}$$

where for each subject i

$$\begin{aligned}&x_1 \sim N(0,1), \quad x_2 \sim \mathrm{Un}(1,2), \\ &x_3 \sim \mathrm{Un}(2,3), \quad x_4 \sim N(0,1), \\ &x_l \overset{i.i.d.}{\sim} N(0,1), \quad l = 5, \ldots, p, \text{ are unrelated feature.}\end{aligned}$$

For each subject i, time values t_{ij} for $j = 1, \ldots, n_i$ are sampled with replacement from $\{1/3, 2/3, \ldots, 3\}$, where the number of time points n_i is drawn randomly from $\{1, \ldots, 9\}$.

Model 2 for baseline covariates:

$$\begin{aligned}\mu_{ij}(\mathbf{x}_i^*, t_{ij}) = 1.5 + 2x_{i3} + 0.5x_{i4}^2 \\ + I_{(t_{ij} \leq 2)}1.5x_{i1} + I_{(t_{ij} > 2)}1.2x_{i2},\end{aligned}$$

where for each subject i

$$
\begin{aligned}
&x_1 \sim \mathrm{Un}(1,3), \quad x_2 \sim \mathrm{N}(2,1),\\
&x_3 \sim \mathrm{N}(0,4), \quad x_4 \sim \mathrm{Un}(0,2),\\
&x_l \overset{i.i.d.}{\sim} N(0,1), \quad l = 5,\dots,p, \text{ are unrelated features.}
\end{aligned}
$$

For each subject i, time values t_{ij} for $j = 1,\dots,n_i$ are sampled from $\mathrm{Un}(0,6)$ where the number of time points n_i is drawn randomly from $\{15,16,17\}$.

Scenario 3. In this scenario, $\mathbf{x}$ is a time-varying covariate vector, which is more common in practical applications. This means that $\mathbf{y}_i$ changes not only with t, but also with the repeated measurements of $\mathbf{x}$, making the model more flexible and complex. Further differences can be seen by comparing the performance of these methods under Scenario 2 and Scenario 3. Common models include time-varying coefficient models and nonlinear mixed effects models. For each individual i, n_i is drawn randomly from $\{3,4,\dots 10\}$, and time t_{ij} for are sampled from $\mathrm{Un}(0,6)$.

Model 1 as the varying coefficient model:

$$
\mu_i\left(\mathbf{x}_{ij}, t_{ij}\right) = \sum_{l=1}^{4} \beta_l\left(t_{ij}\right) x_{ijl}\left(t_{ij}\right),
$$

where for each subject i

$$
\begin{aligned}
&x_1(t) = 1, \quad x_2(t) \sim \mathrm{Bern}(0.6), \quad x_3(t) \sim \mathrm{Un}(0.1t, 2+0.1t),\\
&x_4(t) \mid x_3(t) \sim N\left(0, \frac{1+x_3(t)}{2+x_3(t)}\right),\\
&x_l(t) \overset{i.i.d.}{\sim} N(0,4), \quad l = 5,\dots,p, \text{ are unrelated feature}\\
&\beta_1(t) = 1 + 3.5\sin(t-3), \quad \beta_2(t) = 2 - 5\cos(0.75t - 0.25),\\
&\beta_3(t) = 4 - 0.04(t-12)^2, \quad \beta_4(t) = 1 + 0.125t + 4.6(1-0.1t)^3.
\end{aligned}
$$

Model 2 for time-varying covariates:

$$
\mu_{ij}(\mathbf{x}_{ij}, t_{ij}) = 2x_{ij1}^{0.5} + 1.3x_{ij2}^2 + 5sin(x_{ij3} + x_{ij4}),
$$

where for each subject i

$$
\begin{aligned}
&x_1(t) \sim \mathrm{Un}(0.05 + 0.1t, 2.05 + 0.1t),\\
&x_2(t) \sim N(3\exp\{(t+0.5)/30\}, 1),\\
&x_3(t) \sim \mathrm{Un}(2,4) - 3\cos\{\pi(t-24.5)/15\},\\
&x_4 \sim N(0, 0.1t^2),\\
&x_j(t) \overset{i.i.d.}{\sim} N(0,4), \quad j = 5,\dots,p, \text{ are unrelated feature.}
\end{aligned}
$$

Model 3 for time-varying covariates:

$$
\begin{aligned}
\mu_{ij}(\mathbf{x}_{ij}, t_{ij}) =& 2x_{ij1}^{0.5} + 1.3x_{ij2}^2 + I_{(t_{ij}<2.5)} 5sin(x_{ij3} + x_{ij4})\\
&+ I_{(t_{ij}>3)} exp(0.5x_{ij1}),
\end{aligned}
$$

where the distribution of $\mathbf{x}_{ij}$ are the same as Model 2 with time-varying covariates.

The prediction errors are calculated based on the new measurements of the training individuals. For the REEM tree and VCM methods, we implement the procedures introduced by Capitaine et al. [6] and Wang et al. [57], respectively. For GLMMtree, we let $\mathbf{s}^{sp} = \mathbf{s}^{rg} = \mathbf{x}$. Due to the rapidly increasing computational burden of RETCO with increasing depth, we have restricted the maximum depth to 3. Therefore, the performance of RETCO is expected to be poor.

4.3 Simulation outcomes

Since the FPCA method performs poorly when m is small, we only list the results of $m = 10$ and $m = 15$. In addition, as FPCA is not based on traditional longitudinal data models, we did not assume the working matrix for it. Table 1 compares the efficiency of using the true covariance relative to the independence matrix for each method, and we can draw the following conclusions:

- There is little improvement in efficiency when using the true covariance in kernel GEE, but the remaining methods reduce the MAE when the matrix is correctly specified. This improvement increases as the correlation becomes stronger. This provides evidence that incorporating the within-subject correlation structure into the estimation procedure is crucial for longitudinal data analysis.
- In general, the SUR and smoothing regression methods perform the best among all the methods in different types of models. The SUR method further utilizes the information in $\mathbf{V}_i$ compared to the kernel GEE method. The smoothing spline method uses more knots than the regression spline method to fit flexible and complex models and adds smoothness penalties to improve generalization performance. However, the cost of both methods is a significant increase in computation time.
- Additionally, even in a one-dimensional case, when the sample size is large enough, the performance of the random forest weight method can be on par with that of traditional kernel methods. Therefore, the traditional kernel method may benefit from further development by incorporating the advantages of random forest in high-dimensional data. We also attempted to replace the kernel function with the generalized random forest (GRF) weight used in Athey et al. [1], but the performance is very poor, indicating that the choice of a suitable random forest is also a topic worth exploring.
- Although improving the utilization of the working matrix, the LLME with the true covariance matrix, does not demonstrate advantages over kernel GEE in the simulation, since the random effects can not be identified to help predict. FPCA is derived from the study of functional data; thus, it may lack competitiveness in the context of usual longitudinal data. However, it has demonstrated a certain level of robustness to different

Table 1. Simulation Result for the univariate case under Scenario 1

		Average of MAE using true				Average of MAE ratio using true versus ID			
Setting	Method	M1	M2	M3	M4	M1	M2	M3	M4
	KGEE	0.1541	0.1803	0.2147	0.1757	0.9907	0.9820	0.9904	0.9821
N=50	SUR	0.1412	**0.1646**	**0.1941**	0.1640	0.8897	**0.8698**	**0.8432**	**0.9017**
m=3	SUR-RFW	0.1675	0.2014	0.2975	0.1701	0.9857	0.9886	0.9530	0.9356
$\rho = 0.6$	R-spline	0.1614	0.1875	0.3152	0.1976	**0.8638**	0.8732	0.9572	0.9250
	S-spline	**0.1380**	0.1795	0.2619	**0.1633**	0.8822	0.9097	0.8750	0.9122
	LLME	0.1567	0.2019	0.3238	0.1802	0.9054	0.9457	0.8576	0.9363
	KGEE	0.1118	0.1385	0.1693	0.1400	0.9556	0.9882	0.9912	0.9806
N=100	SUR	0.1047	**0.1238**	**0.1567**	0.1299	0.8965	0.8799	0.9183	0.8972
m=3	SUR-RFW	0.1089	0.1338	0.1735	**0.1247**	0.8457	0.9042	0.9090	**0.8670**
$\rho = 0.6$	R-spline	0.1201	0.1412	0.2794	0.1572	0.8598	0.8778	0.9902	0.8854
	S-spline	**0.0997**	0.1400	0.2112	0.1291	**0.8183**	**0.8670**	0.8693	0.8959
	LLME	0.1251	0.1798	0.3021	0.1469	0.8632	0.9123	**0.8482**	0.8997
	KGEE	0.0975	0.1167	0.1485	0.1224	0.9801	0.9776	0.9787	0.9876
N=150	SUR	0.0873	**0.1087**	0.1400	**0.1112**	0.8491	0.8844	0.9225	0.8724
m=3	SUR-RFW	0.0970	0.1117	**0.1343**	0.1147	**0.8378**	**0.8554**	0.8973	**0.8626**
$\rho = 0.6$	R-spline	0.1022	0.1235	0.2666	0.1421	0.8682	0.8997	0.9941	0.9180
	S-spline	**0.0872**	0.1229	0.1848	0.1149	0.8915	0.8663	0.8623	0.8931
	LLME	0.1106	0.1652	0.2822	0.1359	0.8379	0.8935	**0.8361**	0.9018
	KGEE	0.1479	0.1769	0.2134	0.1709	0.9679	1.0013	0.9805	0.9656
N=50	SUR	0.1454	**0.1729**	**0.2094**	0.1680	0.9849	**0.9579**	0.9654	**0.9450**
m=3	SUR-RFW	0.1605	0.2024	0.2985	0.1709	0.9575	0.9978	0.9901	0.9781
$\rho = 0.3$	R-spline	0.1679	0.1962	0.3208	0.2005	**0.9348**	0.9610	0.9835	0.9748
	S-spline	**0.1404**	0.1877	0.2731	**0.1665**	0.9438	0.9693	0.9513	0.9665
	LLME	0.1588	0.2045	0.3346	0.1779	0.9835	0.9733	**0.9215**	0.9759
	KGEE	0.1558	0.1826	0.2212	0.1810	1.0032	0.9771	0.9874	0.9895
N=50	SUR	0.1307	**0.1547**	**0.1845**	**0.1541**	0.7899	0.7915	0.7908	0.8287
m=3	SUR-RFW	0.1602	0.1961	0.2937	0.1694	0.8758	0.9150	0.9375	0.9155
$\rho = 0.8$	R-spline	0.1472	0.1772	0.3201	0.1894	**0.7374**	0.7955	0.9744	0.8676
	S-spline	**0.1272**	0.1627	0.2382	0.1543	0.7706	**0.7762**	**0.7622**	**0.8103**
	LLME	0.1574	0.1992	0.3170	0.1799	0.8771	0.9087	0.8265	0.9372
	KGEE	0.1015	0.1215	0.1502	0.1266	0.9655	1.0015	0.9763	0.9782
N=50	SUR	0.0925	**0.1098**	0.1452	**0.1155**	0.8447	0.8542	0.9020	**0.8521**
m=10	SUR-RFW	0.1031	0.1146	**0.1342**	0.1208	0.8281	0.8678	0.8764	0.8532
$\rho = 0.6$	R-spline	0.1055	0.1304	0.2861	0.1493	0.8807	0.9562	1.1003	0.9544
	S-spline	**0.0910**	0.1246	0.1839	0.1163	0.8701	**0.8500**	0.8506	0.8550
	LLME	0.1200	0.1707	0.2939	0.1406	**0.8038**	0.8712	**0.8071**	0.8746
	FPCA	0.1498	0.1513	0.1555	0.1585	*	*	*	*
	KGEE	0.0912	0.1074	0.1381	0.1152	0.9924	1.0117	0.9750	0.9934
N=50	SUR	0.0821	**0.0993**	0.1343	**0.1042**	0.9009	0.8854	0.9181	0.8918
m=15	SUR-RFW	0.1007	0.1060	**0.1145**	0.1124	0.8125	**0.8084**	0.8673	**0.8257**
$\rho = 0.6$	R-spline	0.0927	0.1252	0.2839	0.1450	0.8477	1.0205	1.1379	1.0321
	S-spline	**0.0795**	0.1088	0.1649	0.1072	0.8480	0.8416	0.8361	0.8606
	LLME	0.1128	0.1668	0.2865	0.1358	**0.7993**	0.8604	**0.7837**	0.8763
	FPCA	0.1306	0.1298	0.1368	0.1382	*	*	*	*

Note: the FPCA method performs poorly when m is small, we only list the results for $m = 10$ and $m = 15$. Additionally, we do not assume the true working matrix for FPCA, so only compare its practical MAE and denote the MAE ratio with an asterisk (*). Values displayed in bold indicate the winning method for each experiment.

models, making it a preferred choice when the underlying model is complex and the data is dense.

In Scenario 2. we consider **x** to be time-invariant covariates and it is necessary to mention that both GLMM tree and VCM methods may encounter challenges in solving singular matrices in this scenario. In such situations, we only consider the cases from the non-singular portions. It can be observed in Table 2:

- Splinetree does not confer any advantages for baseline covariates. In Model 1, insufficient repeated measurements might result in poor fitting when projecting individual-specific growth curves. Even with an increased size of n_i in Model 2, the performance remains unsatisfactory. This may indicate that splinetree has poor generalization ability due to its neglect of internal correlations within individuals.
- Boostmtree achieves the lowest SMAE in Model 1, but the SMAE of the piecewise function increases significantly compared to the continuous case. This may be due to the increased complexity of the interaction between the variable t and $\mathbf{x}^*$, making gradient approximation more difficult during the boosting process. Even with an increase in sample size, there is no observed improvement in the results.
- The historical tree degenerates into a traditional random forest for baseline covariate and performs much better in the continuous case than the piecewise one.
- The potential model dramatically influences the performance of the REEM tree and GLMMtree. When the sample size increases, both of them provide better performance. In this scenario, the REEM tree always performs better than GLMMtree.
- VCM shows robustness to changes in p while it is affected by the smoothness of the latent model.

In Scenario 3. we consider **x** as time-varying covariates and it can be found in Table 3:

- When many time-varying covariates are present, the performance of the historical tree declines. This may be attributed to the influence of noisy features and the summary function's inability to effectively capture the longitudinal data structure.
- Similarly, we observe that the model, sample size, and dimensionality have a significant impact on the performance of GLMMtree and REEM tree. However, when there are enough observations, the GLMMtree consistently outperforms the REEM tree. In cases where p is large relative to the sample size, the REEM tree may be a better choice to handle the data.
- The VCM method is still barely impacted by changes in p. The model complexity and sample size have less influence on it. This may be attributed to the adaptive selection of predictor-specific knots employed by Wang et al. [57] when constructing basic splines to fit the coefficients in varying coefficient models.
- Due to the small number of nodes generated (no more than 8), RETCO performs poorly in predicting continuous fixed effects. In addition, in the simulation, it is found that RETCO can only identify strong signals. For example, in model 2, it mostly only splits based on x_2.

Table 2. Simulation Result for baseline covariates under Scenario 2

		SMAE					
		M1			M2		
		N=50	N=100	N=150	N=50	N=100	N=150
p=5	Splinetree	0.4598	0.4451	0.3745	0.6367	0.5761	0.5600
	Boostmtree	**0.1661**	**0.1568**	0.1564	0.3550	0.3529	0.3638
	historical tree	0.3737	0.3274	0.3019	0.4565	0.4411	0.4375
	REEM	0.2381	0.2073	0.1923	**0.3400**	**0.3387**	0.3385
	VCM	0.2134	0.2147	0.2102	0.4559	0.4574	0.4564
	GLMMtree	0.2186	0.1596	**0.1483**	0.3488	0.3402	**0.3333**
p=15	Splinetree	0.4619	0.4481	0.3755	0.6003	0.5253	0.5056
	Boostmtree	**0.1713**	**0.1568**	**0.1543**	0.3590	0.3483	0.3522
	historical tree	0.4058	0.3487	0.3261	0.4682	0.4515	0.4482
	REEM	0.2567	0.2249	0.2096	**0.3415**	**0.3408**	**0.3394**
	VCM	0.2245	0.2107	0.2110	0.4570	0.4489	0.4458
	GLMMtree	0.3249	0.2321	0.2021	0.3512	0.3433	0.3453
p=30	Splinetree	0.4618	0.4469	0.3795	0.5939	0.5165	0.4949
	Boostmtree	**0.1789**	**0.1580**	**0.1549**	0.3774	0.3667	0.3565
	historical tree	0.4417	0.3944	0.3676	0.4861	0.4632	0.4627
	REEM	0.2586	0.2282	0.2113	**0.3462**	**0.3454**	**0.3367**
	VCM	0.2206	0.2070	0.2095	0.4530	0.4499	0.4479
	GLMMtree	0.3398	0.3269	0.2413	0.3517	0.3497	0.3452

Note: Values displayed in bold indicate the winning method for each experiment.

Table 3. Simulation Result for varying covariates under Scenario 3

		SMAE								
		M1			M2			M3		
		N=50	N=100	N=150	N=50	N=100	N=150	N=50	N=100	N=150
p=5	historical tree	0.4537	0.3960	0.3794	0.4133	0.3685	0.3483	0.3757	0.3371	0.3194
	REEM	0.3087	0.2815	0.2707	0.3244	0.2941	0.2772	0.2905	0.2637	0.2432
	VCM	0.3461	0.3345	0.3302	0.3461	0.3341	0.3338	0.3017	0.2947	0.2919
	GLMMtree	**0.2563**	**0.2256**	**0.2230**	**0.2896**	**0.2572**	**0.2468**	**0.2358**	**0.2129**	**0.2081**
	RETCO	0.5626	0.5505	0.5422	0.5162	0.5258	0.5062	0.4732	0.4690	0.4476
p=15	historical tree	0.4960	0.4497	0.4193	0.4483	0.4051	0.3785	0.4125	0.3672	0.3466
	REEM	0.3485	0.3030	0.2871	0.3645	0.3291	0.3102	0.3174	0.2884	0.2693
	VCM	**0.3483**	0.3356	0.3323	**0.3323**	0.3399	0.3371	0.2888	0.2886	0.2883
	GLMMtree	0.3897	**0.2840**	**0.2308**	0.3484	**0.2959**	**0.2739**	**0.2811**	**0.2419**	**0.2148**
	RETCO	0.5679	0.5525	0.5444	0.5285	0.5148	0.5055	0.4786	0.4617	0.4644
p=30	historical tree	0.5884	0.5270	0.4984	0.5262	0.4769	0.4436	0.4891	0.4326	0.4130
	REEM	0.3652	**0.3211**	0.2972	0.3900	0.3435	0.3226	0.3295	0.2890	0.2735
	VCM	**0.3437**	0.3380	0.3309	**0.3493**	**0.3361**	0.3338	0.3066	0.2915	0.2904
	GLMMtree	0.4027	0.3872	**0.2933**	0.3616	0.3454	**0.3125**	**0.2872**	**0.2803**	**0.2475**
	RETCO	0.5672	0.5534	0.5483	0.5395	0.5245	0.5091	0.5001	0.4879	0.4603

Note: Values displayed in bold indicate the winning method for each experiment.

This may be attributed to the new stopping rule. While such a dilemma may potentially be solved by increasing the depth, the computation time required for this approach would be significantly longer than that of other methods.

- Comparing the SMAE in Model 2 from Scenario 2 and Model 3 in Scenario 3, we observe that even with similar models, differences in the varying covariates and baseline covariates can have a significant impact on prediction errors. This is mainly because it affects whether the repeated measurements belonging to the same individual will be split into different child nodes when the node is partitioned on $\mathbf{x}$.

In general, when $\mathbf{x}^*$ are baseline covariates, boostmtree and REEM tree are recommended for smooth signals. However, when $\mathbf{x}$ are time-varying covariates, the GLMMtree may perform better when N is large. In high-dimensional cases, the REEM tree is expected to be more appropriate. In any case, VCM shows its robustness.

5. DISCUSSIONS

In this paper, we present a selective review of nonparametric methods for longitudinal data analysis. Excluding the classical univariate methods we only focus on the major developments in combining regression trees with longitudinal data models. These methods have evolved from different models and possess unique features. The marginal model assumes fewer restrictions than the LMM model, which is more commonly used and includes an analysis of random effects. The VCM model is the most extensive but requires a more complex estimation process. While the methods that only consider the influence of univariate time t on the response variable have been widely studied in the past, the SUR method and smoothing spline have shown the best performance in our review, although at the cost of higher computational complexity. Recent technological advances highlight the need to include more covariates related to the response variable to investigate their relationships. To this end, the use of regression trees for longitudinal data analysis has become a popular trend. In Scenarios 2 and 3, we briefly compare their performances under classical models. The outcomes show that there is no one method that uniformly outperforms the others in practice, therefore it is necessary to select the appropriate method based on the specific background. Although some methods showed suboptimal performance in our simulations, we emphasize that this paper primarily focuses on reviewing the motivations and ideas behind different methods, and our simulation results are provided for reference only.

In addition to the reviewed methods, there have also been advancements in parametric and semiparametric models in recent years. Wang et al. [58] extended the GEE method to high-dimensional cases, and Kamruzzaman et al. [25] generalized the response variable of the GEE to a vector based on their ideas. In semiparametric models, Gottard et al. [17] used regression trees to capture non-linearities while retaining the linear part of the LMM. Model-based methods, such as boosting in Hothorn et al. [21] for additive models and MOB in Zeileis et al. [66] used in GLMM trees, can also be employed to combine various parametric and nonparametric models for longitudinal data analysis.

For future directions in methodology development, the first idea is to extend the work in Segal [45] by finding a criterion that enables splitting on time-varying covariates in recognition of the flexible relationship between the response and the covariates when the repeated measurements of y or their projection are used as a vector response variable. This would allow for the extension of many existing methods to a more general case. For example, in the simulation, boostmtree shows competitiveness when it comes to baseline covariates, but its requirements for $\mathbf{x}^*$ limit its practical applications. It is a challenging task to preserve its superiority in baseline covariates and its ability to partition on varying covariates.

Secondly, we suggest exploring the use of different types of machine learning and deep learning methods to estimate fixed effects and even random effects in LMM, building on the spirit of methods like the REEM tree. Different machine learning methods have their own advantages, allowing us to make more appropriate choices under different models. Currently, although Mandel et al. [34] have started using neural networks, similar work is still insufficient. Furthermore, it is known that inaccurate specification of random effects may adversely impact the precision of both REEM trees and GLMM trees. We can use some nonparametric methods to estimate them, such as Capitaine et al. [6], which incorporated a random process $\omega_i(t_{ij})$ in addition to linear random effects. This idea can be further generalized.

The third direction is the local estimation. This method may present two challenges. The first challenge is to determine an appropriate local estimation method, especially when the sample size is small compared to the dimensionality. In the GLMM tree, a linear model is fitted within each node, but when the sample size is not large enough, correctly specifying fixed-effects predictor variables becomes important. Recently, Friedberg et al. [15] presented the local linear forest, which fits a ridge regression within each node to remove the influence of noise features in high-dimensional cases, providing a potential solution to this challenge. The second challenge is selecting an appropriate impurity measure for best splitting in such cases. The parameter model is now commonly used for local estimation due to its extensively researched robustness test of parameter estimators. Additionally, the loss function for correlated data in RETCO provides another tool.

Another purpose of this paper is to investigate the feasibility of combining the kernel method for longitudinal data with random forest weights. While some articles have applied random forest weights to other problems, such as Qiu et al. [38], who replaced the kernel function in traditional Fréchet regression with random forest weights and provided theoretical analysis, and Friedberg et al. [15], who used it for smoothing signals, more work needs to be done to apply random forest weights to longitudinal data analysis. Therefore, our forthcoming investigations will concentrate on this direction.

ACKNOWLEDGEMENTS

The authors would like to thank the Editor and the reviewers for their helpful comments and suggestions on previous versions of the paper. This paper is motivated by the special issue in honor of Professor Lincheng Zhao and pays tribute to his illustrious career and long-term service. This work is partially supported by the National Natural Science Foundation of China grants 12071087, 12226308.

Received 30 December 2022

REFERENCES

[1] Athey, S., Tibshirani, J. and Wager, S. (2019). Generalized random forests. *The Annals of Statistics* **47** 1148–1178. MR3909963

[2] Boente, G., Salibián Barrera, M. and Tyler, D. E. (2014). A characterization of elliptical distributions and some optimality properties of principal components for functional data. *Journal of Multivariate Analysis* **131** 254–264. MR3252648

[3] Breiman, L. (2001). Random Forests. *Machine Learning* **45** 5–32. MR3874153

[4] Bürgin, R. and Ritschard, G. (2015). Tree-based varying coefficient regression for longitudinal ordinal responses. *Computational Statistics & Data Analysis* **86** 65–80. MR3312738

[5] Calhoun, P., Levine, R. A. and Fan, J. (2020). Repeated measures random forests (RMRF): Identifying factors associated with nocturnal hypoglycemia. *Biometrics* **77** 343–351. MR4229744

[6] Capitaine, L., Genuer, R. and Thi'ebaut, R. (2019). Random forests for high-dimensional longitudinal data. *Statistical Methods in Medical Research* **30** 166–184. MR4216853

[7] Cevid, D., Michel, L., Meinshausen, N. and Bühlmann, P. (2022). Distributional Random Forests: Heterogeneity Adjustment and Multivariate Distributional Regression. *Journal of Machine Learning Research* **23** 1–79. MR4577772

[8] Deshpande, S. K., Bai, R., Balocchi, C., Starling, J. E. and Weiss, J. (2020). VCBART: Bayesian trees for varying coefficients. ArXiv: 2003.06416.

[9] Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science* **11** 89–121. MR1435485

[10] Fan, J. and Wu, Y. (2008). Semiparametric Estimation of Covariance Matrixes for Longitudinal Data. *Journal of the American Statistical Association* **103** 1520–1533. MR2504201

[11] Fan, J. and Zhang, W. (1999). Statistical estimation in varying coefficient models. *The Annals of Statistics* **27** 1491–1518. MR1742497

[12] Fan, J. and Zhang, W. (2008). Statistical Methods with Varying Coefficient Models. *Statistics and its interface* **1** 179–195. MR2425354

[13] Fitzmaurice, G., Davidian, M., Verbeke, G. and Molenberghs, G. (2008). *Longitudinal Data Analysis (Chapman & Hall/CRC Handbooks of Modern Statistical Methods)*, 1 ed. Chapman & Hall/CRC. MR1500110

[14] Fokkema, M., Smits, N., Zeileis, A., Hothorn, T. and Kelderman, H. (2018). Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior Research Methods* **50** 2016–2034.

[15] Friedberg, R., Tibshirani, J., Athey, S. and Wager, S. (2018). Local Linear Forests. *Journal of Computational and Graphical Statistics* **30** 503–517. MR4270520

[16] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics* **29** 1189–1232. MR1873328

[17] Gottard, A., Vannucci, G., Grilli, L. and et al. (2023). Mixed-effect models with trees. *Adv Data Anal Classif* **17** 431–461. MR4589700

[18] Green, P. and Silverman, B. (1993). *Nonparametric Regression and Generalized Linear Models: A roughness penalty approach* , 1 ed. Chapman and Hall/CRC. MR1270012

[19] Hajjem, A., Bellavance, F. and Larocque, D. (2011). Mixed effects regression trees for clustered data. *Statistics & Probability Letters* **81** 451–459. MR2765165

[20] Hoover, D. R., Rice, J. A., Wu, C. O. and Yang, L.-P. (1998). Nonparametric Smoothing Estimates of Time-Varying Coefficient Models with Longitudinal Data. *Biometrika* **85** 809–822. MR1666699

[21] Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M. and Hofner, B. (2010). Model-based Boosting 2.0. *Journal of Machine Learning Research* **11** 2109–2113. MR2719848

[22] Hu, J. and Szymczak, S. (2022). A review on longitudinal data analysis with random forest in precision medicine. ArXiv: 2208.04112.

[23] Huang, J., Wu, C. and Zhou, L. (2004). Polynomial spline estimation and inference for varying coefficient models with longitudinal data. *Statistica Sinica* **14** 763–788. MR2087972

[24] Ishwaran, H. and Kogalur, U. (2016). randomForestSRC: Random Forests for Survival, Regression and Classification (RF-SRC). *R package version 3.1.1*.

[25] Kamruzzaman, M., Kwon, O. and Park, T. (2021). Penalized generalized estimating equations approach to longitudinal data with multinomial responses. *Journal of the Korean Statistical Society* **50** 844–859. MR4303531

[26] Laird, N. M. and Ware, J. H. (1982). Random-Effects Models for Longitudinal Data. *Biometrics* **38** 963–974.

[27] Lee, S.-K. (2005). On generalized multivariate decision tree by using GEE. *Comput. Stat. Data Anal.* **49** 1105–1119. MR2143060

[28] Lee, S.-K. (2006). On Classification and Regression Trees for Multiple Responses and Its Application. *Journal of Classification* **23** 123–141. MR2256202

[29] Liang, K.-Y. and Zeger, S. L. (1986). Longitudinal Data Analysis Using Generalized Linear Models. *Biometrika* **73** 13–22. MR0836430

[30] Lin, X. and Carroll, R. J. (2000). Nonparametric Function Estimation for Clustered Data When the Predictor is Measured without/with Error. *Journal of the American Statistical Association* **95** 520–534. MR1803170

[31] Lin, X. and Carroll, R. J. (2006). Semiparametric Estimation in General Repeated Measures Problems. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **68** 69–88. MR2212575

[32] Lin, X., Wang, N., Welsh, A. H. and Carroll, R. J. (2004). Equivalent kernels of smoothing splines in nonparametric regression for clustered/longitudinal data. *Biometrika* **91** 177–193. MR2050468

[33] Liu, X. (2016). Chapter 1 - Introduction. In *Methods and Applications of Longitudinal Data Analysis* (X. Liu, ed.) 1-18. Academic Press, Oxford.

[34] Mandel, F., Ghosh, R. P. and Barnett, I. (2022). Neural networks for clustered and longitudinal data using mixed effects models. *Biometrics* **00** 1–11.

[35] Neufeld, A. (2019). splinetree: Regression Spline Functions and Classes. *R package version 0.2*.

[36] Pande, A., Ishwaran, H. and Blackstone, E. H. (2022). Boosting for Multivariate Longitudinal Responses. *SN Comput. Sci.* **3** 186.

[37] Pande, A., Li, L., Rajeswaran, J., Ehrlinger, J., Kogalur, U. B., Blackstone, E. H. and Ishwaran, H. (2017). Boosted Multivariate Trees for Longitudinal Data. *Machine Learning* **106** 277–305. MR3596880

[38] Qiu, R., Yu, Z. and Zhu, R. (2022). Random Forests Weighted Local Fréchet Regression with Theoretical Guarantee. ArXiv: 2202.04912.

[39] Rabinowicz, A. and Rosset, S. (2019). Cross-Validation for Correlated Data. *Journal of the American Statistical Association* **117** 718–731. MR4436308

[40] Rabinowicz, A. and Rosset, S. (2022). Tree-Based Models for Correlated Data. *Journal of Machine Learning Research* **23** 1–31. MR4577697

[41] Rice, J. A. and Silverman, B. W. (1991). Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal of the Royal Statistical Society Series b-methodological* **53** 233–243. MR1094283

[42] Rice, J. A. and Wu, C. (2001). Nonparametric Mixed Effects Models for Unequally Sampled Noisy Curves. *Biometrics* **57** 253–259. MR1833314

[43] Ruppert, D. (1997). Empirical-Bias Bandwidths for Local Polynomial Nonparametric Regression and Density Estimation. *Journal of the American Statistical Association* **92** 1049–1062. MR1482136

[44] Ruppert, D. and Wand, M. P. (1994). Multivariate Locally Weighted Least Squares Regression. *The Annals of Statistics* **22** 1346–1370. MR1311979

[45] Segal, M. R. (1992). Tree-Structured Methods for Longitudinal Data. *Journal of the American Statistical Association* **87** 407–418.

[46] Sela, R. and Simonoff, J. (2012). RE-EM trees: A data mining approach for longitudinal and clustered data. *Machine Learning* **86** 169–207. MR2892116

[47] Sexton, J. and Laake, P. (2018). htree: Historical Tree Ensembles for Longitudinal Data. *R package version 2.0.0*.

[48] Shi, M., Weiss, R. E. and Taylor, J. M. G. (1996). An Analysis of Paediatric CD4 Counts for Acquired Immune Deficiency Syndrome Using Flexible Random Curves. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **45** 151–163.

[49] Sigrist, F. (2020). Gaussian Process Boosting. ArXiv: 2004.02653. MR4577671

[50] Sigrist, F. (2021). Latent Gaussian Model Boosting. ArXiv: 2105.08966. MR4577671

[51] Stone, C. J., Hansen, M. H., Kooperberg, C. L. and Truong, Y. (1997). Polynomial splines and their tensor products in extended linear modeling: 1994 Wald memorial lecture. *Annals of Statistics* **25** 1371–1470. MR1463561

[52] Sun, Y., Zhang, W. and Tong, H. (2007). Estimation of the covariance matrix of random effects in longitudinal studies. *The Annals of Statistics* **35** 2795–2814. MR2382666

[53] Wand, M. P. (2000). A Comparison of Regression Spline Smoothing Procedures. *Computational Statistics* **15** 443–462. MR1818029

[54] Wang, N. (2003). Marginal Nonparametric Kernel Regression Accounting for Within-Subject Correlation. *Biometrika* **90** 43–52. MR1966549

[55] Wang, N., Carroll, R. J. and Lin, X. (2005). Efficient Semiparametric Marginal Estimation for Longitudinal/Clustered Data. *Journal of the American Statistical Association* **100** 147–157. MR2156825

[56] Wang, J. C. and Hastie, T. J. (2014). Boosted Varying-Coefficient Regression Models for Product Demand Prediction. *Journal of Computational and Graphical Statistics* **23** 361–382. MR3215815

[57] Wang, X., Jiang, B. and Liu, J. S. (2022). Varying Coefficient Model via Adaptive Spline Fitting. ArXiv: 2201.10063.

[58] Wang, L., Zhou, J. and Qu, A. (2012). Penalized generalized estimating equations for high-dimensional longitudinal data analysis. *Biometrics* **68** 353–360. MR2959601

[59] Wu, C. O., Chiang, C.-T. and Hoover, D. R. (1998). Asymptotic Confidence Regions for Kernel Smoothing of a Varying-Coefficient Model with Longitudinal Data. *Journal of the American Statistical Association* **93** 1388–1402. MR1666635

[60] Wu, C. O. and Chiang, C.-T. (2000). Kernel smoothing on varying coefficient models with longitudinal dependent variable. *Statistica Sinica* **10** 433–456. MR1769751

[61] Wu, H. and Zhang, J. (2002). Local Polynomial Mixed-Effects Models for Longitudinal Data. *Journal of the American Statistical Association* **97** 883–897.

[62] Wu, H. and Zhang, J. (2006). *Nonparametric Regression Methods for Longitudinal Data Analysis: Mixed-Effects Modeling Approaches*, 1 ed. Wiley-Interscience. MR2216899

[63] YAO, F., MÜLLER, H.-G. and WANG, J.-L. (2005). Functional Data Analysis for Sparse Longitudinal Data. *Journal of the American Statistical Association* **100** 577–590. MR2160561

[64] YU, Y. and LAMBERT, D. (1999). Fitting Trees to Functional Data, with an Application to Time-of-Day Patterns. *Journal of Computational and Graphical Statistics* **8** 749–762.

[65] YUE, M., LI, J. and CHENG, M.-Y. (2019). Two-step sparse boosting for high-dimensional longitudinal data with varying coefficients. *Comput. Stat. Data Anal.* **131** 222–234. MR3906806

[66] ZEILEIS, A., HOTHORN, T. and HORNIK, K. (2008). Model-Based Recursive Partitioning. *Journal of Computational and Graphical Statistics* **17** 492–514. MR2439970

[67] ZHOU, Y. and HOOKER, G. (2022). Decision tree boosted varying coefficient models. *Data Mining and Knowledge Discovery* **36** 2237–2271. MR4510524

[68] ZHU, Z., FUNG, W. K. and HE, X. (2008). On the asymptotics of marginal regression splines with longitudinal data. *Biometrika* **95** 907–917. MR2461219

Changxin Yang
Department of Statistics and Data Science
Fudan University
Shanghai
China
E-mail address: yangcx22@m.fudan.edu.cn

Zhongyi Zhu
Department of Statistics and Data Science
Fudan University
Shanghai
China
E-mail address: zhuzy@fudan.edu.cn