Longitudinal Deep Kernel Gaussian Process Regression

Junjie Liang, Yanting Wu, Dongkuan Xu, Vasant Honavar
Pennsylvania State University
{jul672, yxw514, dux19, vhonavar}@psu.edu

Abstract

Gaussian processes offer an attractive framework for predictive modeling from longitudinal data, *i.e.,* irregularly sampled, sparse observations from a set of individuals over time. However, such methods have two key shortcomings: (i) They rely on ad hoc heuristics or expensive trial and error to choose the effective kernels, and (ii) They fail to handle multilevel correlation structure in the data. We introduce Longitudinal deep kernel Gaussian process regression (L-DKGPR) to overcome these limitations by fully automating the discovery of complex multilevel correlation structure from longitudinal data. Specifically, L-DKGPR eliminates the need for ad hoc heuristics or trial and error using a novel adaptation of deep kernel learning that combines the expressive power of deep neural networks with the flexibility of non-parametric kernel methods. L-DKGPR effectively learns the multilevel correlation with a novel additive kernel that simultaneously accommodates both time-varying and the time-invariant effects. We derive an efficient algorithm to train L-DKGPR using latent space inducing points and variational inference. Results of extensive experiments on several benchmark data sets demonstrate that L-DKGPR significantly outperforms the state-of-the-art longitudinal data analysis (LDA) methods.

Introduction

Longitudinal studies, which involve repeated observations, taken at irregularly spaced time points, for a set of individuals over time, are ubiquitous in many applications, e.g., in health, cognitive, social, and economic sciences. Such studies are used to identify the time-varying as well as the time-invariant factors associated with a particular outcome of interest, *e.g.,* health risk (Hedeker and Gibbons 2006), urban computing (Tang et al. 2020; Hsieh et al. 2021). Longitudinal data typically exhibit longitudinal correlation (LC), *i.e.,* correlations among the repeated observations of a given individual over time; and cluster correlation (CC), *i.e.,* correlations among observations across individuals, *e.g.,* due to the characteristics that they share among themselves e.g., age, demographics factors; or both, *i.e.,* multilevel correlation (MC). In general, the structure of MC can be complex and a priori unknown. Failure to adequately account for the structure of MC in predictive modeling from longitudinal data can lead to misleading statistical inferences (Gibbons and Hedeker 1997; Liang et al. 2020). It can be non-trivial to choose a suitable correlation structure that reflects the correlations present in the data. The relationships between the covariates and outcomes of interest can be highly complex and non-linear. Furthermore, modern applications often call for LDA methods that scale gracefully with increasing number of variables, the number of individuals, and the number of longitudinal observations per individual.

Related Work

Conventional LDA Methods LDA methods have been extensively studied for decades (Hedeker and Gibbons 2006; Verbeke et al. 2014). Conventional LDA methods fall into to two broad categories: (i) marginal models and (ii) conditional models. Marginal models rely on assumptions about the marginal association among the observed outcomes. The generalized estimating equations (GEE) (Liang and Zeger 1986), where a working correlation matrix is specified to model the marginal association among the observed outcomes, offer an example of marginal models. The parameters of marginal models are often shared by all individuals in the population, yielding *population-averaged* effects or *fixed* effects. Conditional models on the other hand avoid directly specifying the full correlation matrix by distinguishing *random* effects, *i.e.,* parameters that differ across individuals, from fixed effects, so as to estimate the individual parameters conditioned on the population parameters. A popular example of conditional models is the generalized linear mixed-effects models (GLMM) (McCulloch 1997). Despite much work on both marginal and conditional models (Fitzmaurice, Laird, and Ware 2012; Wang 2014; Xiong, Kim, and Singh 2019; Liang et al. 2020), many of the challenges, especially the choice of correlation structure, and the selection of variables to model random versus fixed effects, and the scalability of the methods remain to be addressed.

Non-parametric LDA Methods More recently, there is a growing interest in Gaussian processes (GP) (Quintana et al. 2016; Cheng et al. 2019; Wang et al. 2019) for LDA because of their advantages over conventional parametric LDA methods: (i) GP make fewer assumptions about the underlying data distribution by dispensing with the need to choose a particular parametric form of the nonlinear predictive model;

(ii) GP permit the use of parametertized kernels to model the correlation between observed outcomes, to cope with data sampled at irregularly spaced time points, by interpolating between samples; (iii) The interpretability of GP models can be enhanced by choosing modular kernels that are composed of simpler kernels that capture the shared correlation structure of a subset of covariates, and (iv) GP models can flexibly account for both longitudinal and cluster correlations in the data. For example, Cheng et al. (2019) utilize an additive kernel for Gaussian data and employ a step-wise search strategy to select the kernel components and covariates that optimize the predictive accuracy of the model. Timonen et al. (2019) consider a heterogeneous kernel to model individual-specific (random) effects in the case of non-Gaussian data. Despite their advantages, existing GP based approaches to LDA suffer from several shortcomings that limit their applicability in real-world settings: (i) The choice of an appropriate kernel often involves a tedious, often expensive and unreliable, process of trial and error (Rasmussen 2003) or ad hoc heuristics for identifying a kernel or selecting a subset of kernels from a pool of candidates (Cheng et al. 2019). (ii) Suboptimal choice of kernels can fail to adequately model the complex MC structure in the data. (iii) They do not scale to thousands of covariates and/or millions of data points that are common in modern LDA applications.

Overview of contributions

A key challenge in predictive modeling of longitudinal data has to do with modeling the complex correlation structure in the data. We posit that the observed correlation structure is induced by the interactions between time-invariant, individual-specific effects, and time-varying population effects. Hence, we can divide the task of predictive modeling from longitudinal data into three sub-tasks: (i) Given an observed data set, how do we estimate the time-varying and time-invariant effects? (ii) Given the learned effects, how do we estimate the correlation structure present in the data? (iii) Given the correlation structure, how do we predict as yet unobserved, *e.g.,* future outcomes?

We introduce Longitudinal deep kernel Gaussian process regression (L-DKGPR) to fully automate the discovery of complex multi level correlation structure from longitudinal data. L-DKGPR inherits the attractive features of GP while overcoming their key limitations. Specifically, L-DKGPR eliminates the need for ad hoc heuristics or trial and error by using a deep kernel learning method (Wilson et al. 2016a) that combines the expressive power of deep neural networks with the flexibility of non-parametric kernel methods. L-DKGPR extends (Wilson et al. 2016a) by introducing a novel additive kernel that includes two components, one for modeling the time-varying (fixed) effects and the other for modeling the time-invariant (random) effects, to compensate for the multilevel correlation structure in longitudinal data. To enhance the effectiveness and efficiency of model inference, we improve the inducing points technique by introducing inducing points directly in the latent space. Our formulation permits a tractable ELBO, which not only eliminates the need for Monte Carlo sampling, but also dramatically reduces the number of parameters and iterations needed to achieve state-of-the-art regression performance.

Preliminaries

Notations. We denote a longitudinal data set by $\mathcal{D} = (X, \boldsymbol{y})$, where $X \in \mathbb{R}^{N\times P}$ is the covariate matrix and $\boldsymbol{y} \in \mathbb{R}^{N\times 1}$ is the vector of measured outcomes. We denote a row in X by $\boldsymbol{x}_{it}$, with i, t indexing the individual and the time for the observation respectively. Because the observations for each individual are irregularly sampled over time, we have for each individual i, a submatrix $X_i \in \mathbb{R}^{N_i\times P} \subset X$, where N_i is the number of observations available for the individual i. If we denote by I be the number of individuals in $\mathcal{D}$, the covariate matrix X is given by $X^\top = (X_1^\top, \cdots, X_I^\top)^\top$. Accordingly, the outcomes $\boldsymbol{y}$ are given by $\boldsymbol{y}^\top = (\boldsymbol{y}_1^\top, \cdots, \boldsymbol{y}_I^\top)^\top$.

Gaussian Process. A Gaussian process (GP) is a stochastic process, i.e., a distribution over functions or an infinite collection of (real-valued) random variables, such that any finite subset of random variables has a multivariate Gaussian distribution (Williams and Rasmussen 2006). A kernel describes the covariance of the random variables that make up the GP. More precisely, if a function $f : \mathcal{X} \to \mathbb{R}$ has a GP prior $f \sim \mathcal{GP}(\mu, k_\gamma)$ where μ is the mean function and $k_\gamma(\cdot,\cdot)$ is a (positive semi-definite) kernel function parameterized by γ, then any finite collection of components of f (denoted as $\mathbf{f}$) has a multivariate Gaussian distribution $(\mathbf{f}|X) \sim \mathcal{N}(\mu(X), K_{XX})$, where $\mu(X)$ is the mean vector, and $(K_{XX})_{ij} = k_\gamma(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the covariance matrix. In the regression setting, the function f is treated as an unobserved signal linked to the outcomes through a (typically Gaussian) likelihood function, such that $(y|\mathbf{f}) \sim \mathcal{N}(\mathbf{f}, \sigma^2\mathrm{I})$.

Additive GP is a special case of GP where unobserved signal is expressed as the sum of J independent signal components, *i.e.,* $f = \sum_{j=1}^{J} \alpha_j f^{(j)}$, where $\boldsymbol{\alpha} = \{\alpha_j\}_{j=1}^{J}$ are the the coefficients associated with the individual components (Duvenaud, Nickisch, and Rasmussen 2011). In practice, each signal component is computed on a (typically small (Cheng et al. 2019; Timonen et al. 2019)) subset of the observed covariates in $\boldsymbol{x}$. The fact that each signal component has a GP prior ensures that the joint signal f is also GP. Additive GP allows using different kernel functions for different signal components, so to model the shared correlation structure of a subset of covariates, thus enhancing the interpretability of the resulting GP. More importantly, it permits the time-varying and time-invariant effects to be modeled using different kernel functions, which is especially attractive in modeling longitudinal data.

Longitudinal Deep Kernel Gaussian Process Regression

Predictive modeling from longitudinal data typically requires solving two sub-problems: (i) Extracting the time-varying and time-invariant information from the observed data to estimate the underlying multilevel correlation structure; and (ii) using the estimated correlation structure to predict the future outcomes. In what follows, we describe our solutions to both sub-problems.

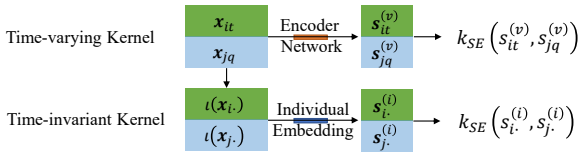

Figure 1: Structure of the deep kernels.

Modeling the Multilevel Correlation using Deep Kernels

Recall that longitudinal data exhibit complex correlations arising from the interaction between time-varying effects and time-invariant effects. Hence, we decompose the signal function f into two parts, *i.e.,* $f^{(v)}$ which models the time-varying effects and $f^{(i)}$, which models the time-invariant effects. The result is a probabilistic model that can be specified as follows:

$$
\begin{aligned}
&(\boldsymbol{y}|\mathbf{f}) \sim \mathcal{N}(\mathbf{f}, \sigma^2 \mathbf{I}) \\
&f = \alpha^{(v)} f^{(v)} + \alpha^{(i)} f^{(i)} \\
&(\mathbf{f}^{(v)}|X) \sim \mathcal{N}(\boldsymbol{\mu}^{(v)}(X), k_\gamma^{(v)}(X,X)) \\
&(\mathbf{f}^{(i)}|X) \sim \mathcal{N}(\boldsymbol{\mu}^{(i)}(X), k_\phi^{(i)}(X,X))
\end{aligned}
$$

We denote the kernel parameters for time-varying effects and time-invariant effects respectively by γ and ϕ. The mean functions $\boldsymbol{\mu}^{(v)}, \boldsymbol{\mu}^{(i)}$, if unknown, can be estimated from data. In this study, without loss of generality, following (Williams and Rasmussen 2006; Wilson et al. 2016a,b; Cheng et al. 2019; Timonen et al. 2019), we set $\boldsymbol{\mu}^{(v)} = \boldsymbol{\mu}^{(i)} = 0$. Assuming that $\mathbf{f}^{(v)}$ and $\mathbf{f}^{(i)}$ are conditionally independent given X, we can express the joint signal distribution $\mathbf{f}$ as follows:

$$
(\mathbf{f}|X) \sim \mathcal{N}\left(\mathbf{0}, k_\theta = {\alpha^{(v)}}^2 k_\gamma^{(v)} + {\alpha^{(i)}}^2 k_\phi^{(i)}\right) \tag{1}
$$

Time-varying Kernel $k_\gamma^{(v)}$. We introduce a time-varying kernel to capture the longitudinal correlation in the data. The structure of our time-varying kernel $k_\gamma^{(v)}$ is shown in the upper part of Figure 1. Let $e_\gamma : \mathcal{X} \to \mathcal{S}^{(v)} \in \mathbb{R}^{D_v}$ be a non-linear encoder function given by a deep architecture parameterized by γ. Given a pair of data points $\boldsymbol{x}_{it}, \boldsymbol{x}_{jq}$, where i, j index the individuals and t, q index the time-dependent observations, the time-varying kernel is given by:

$$
k_\gamma^{(v)}(\boldsymbol{x}_{it}, \boldsymbol{x}_{jq}) = k_{SE}(e_\gamma(\boldsymbol{x}_{it}), e_\gamma(\boldsymbol{x}_{jq})) \tag{2}
$$

with k_{SE} denoting the squared exponential kernel (Williams and Rasmussen 2006). Note that SE kernel is based on Euclidean distance, which is not a useful measure of distance in the high dimensional input space (Aggarwal, Hinneburg, and Keim 2001). Hence, we use a deep neural network (Goodfellow, Bengio, and Courville 2016), specifically, a nonlinear encoder to map the input space to a low-dimensional latent space and then apply the SE kernel to the latent space.

Time-invariant Kernel $k_\phi^{(i)}$. We introduce a time-invariant kernel to capture cluster correlation, *i.e.,* time-invariant correlations among individuals that share similar characteristics. The structure of time-invariant kernel is shown in the bottom part of Figure 1. Let $\iota(\boldsymbol{x}_{i\cdot}) = i$ be a mapping function that identifies the individuals, and $g_\phi : \iota(\mathcal{X}) \to \mathcal{S}^{(i)} \in \mathbb{R}^{D_i}$ be an embedding function that maps each individual to a vector in the latent space. Then for any pair of data points $\boldsymbol{x}_{i\cdot}, \boldsymbol{x}_{j\cdot}$ with arbitrary observation indices, the time-invariant kernel is given by:

$$
k_\phi^{(i)}(\boldsymbol{x}_{i\cdot}, \boldsymbol{x}_{j\cdot}) = k_{SE}(g_\phi \circ \iota(\boldsymbol{x}_{i\cdot}), g_\phi \circ \iota(\boldsymbol{x}_{j\cdot})) \tag{3}
$$

Learning a L-DKGPR model from data

We now proceed to describe how to efficiently learn an L-DKGPR model and use it to make predictions. Because of space constraints, the details of the derivations are relegated to Appendix .

Model Inference. Our approach to efficiently learning an L-DKGPR model draws inspiration from (Wilson et al. 2016b), to greatly simplify the computation of the GP posterior by reducing the effective number of rows in X, from N to M ($M \ll N$), where M is the number of *inducing points*. However, unlike (Wilson et al. 2016b), which uses inducing points in the input space, we use inducing points from *a low-dimensional latent space*. Let $Z = \{\boldsymbol{z}_m\}_{m=1}^M$ be the collection of inducing points, and $\mathbf{u}$ their corresponding signal. The kernel computations based on the inducing points are given by:

$$
\begin{aligned}
k_\gamma^{(v)}(\boldsymbol{x}, \boldsymbol{z}) &= k_{SE}(e_\gamma(\boldsymbol{x}), \boldsymbol{z}) \\
k_\gamma^{(v)}(\boldsymbol{z}_i, \boldsymbol{z}_j) &= k_{SE}(\boldsymbol{z}_i, \boldsymbol{z}_j)
\end{aligned}
$$

Replacing inducing points in the input space with those in a low-dimensional latent space offers several advantages. First, we no longer need to use the encoder network $e_\gamma(\cdot)$ to transform the inducing points $\boldsymbol{z}$, thus increasing the computational efficiency of the model. Second, the latent space is dense, continuous, and usually is of much lower dimension than the input space ($D_v \ll P$). The resulting parameterization of inducing points directly in the latent space, results in a reduction in the number of parameters that describe the inducing points (*i.e., Z*) from $\mathcal{O}(MP)$ to $\mathcal{O}(MD_v)$. Third, the latent space simplifies the optimization of L-DKGPR, especially when the input space is defined by heterogeneous data types subject to domain-specific constraints, because the latent space is always continuous regardless the constraints in the input space. We define $\iota(\boldsymbol{z}_m) = I + m$ to distinguish the inducing points from the input data. We can now express the joint signal distribution as follows:

$$
(\mathbf{f}, \mathbf{u}|X, Z) \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}K_{XX} & K_{XZ}\\ K_{XZ}^\top & K_{ZZ}\end{bmatrix}\right) \tag{4}
$$

Therefore, the signal distribution conditioned on the inducing points is given by:

$$
(\mathbf{f}|\mathbf{u}, X, Z) \sim \mathcal{N}(K_{XZ}K_{ZZ}^{-1}\mathbf{u}, K_{XX} - K_{XZ}K_{ZZ}^{-1}K_{XZ}^\top) \tag{5}
$$

Let $\Theta = \{\alpha^{(v)}, \alpha^{(i)}, \gamma, \phi, \sigma^2, Z\}$ be the model parameters. We aim to learn the parameters by maximizing the log of marginal likelihood $p(\boldsymbol{y}|X, Z)$. By assuming a variational posterior over the joint signals $q(\mathbf{f}, \mathbf{u}|X, Z) =$

$q(\mathbf{u}|X,Z)p(\mathbf{f}|\mathbf{u},X,Z)$, we can derive the evidence lower bound (see *e.g.,* (Wilson et al. 2016b)):

$$\mathcal{L} \triangleq \mathbb{E}_{q(\mathbf{f},\mathbf{u}|X,Z)}[\log p(\boldsymbol{y}|\mathbf{f})] - \mathrm{KL}[q(\mathbf{u}|X,Z)||p(\mathbf{u}|Z)] \quad (6)$$

We define the proposal posterior $q(\mathbf{u}|X,Z) = \mathcal{N}(\boldsymbol{\mu}_q, L_q L_q^\top)$. To speed up the computation, we follow the deterministic training conditional (DTC) (Seeger, Williams, and Lawrence 2003), an elegant sparse method for accurate computation of the Gaussian process posterior by retaining exact likelihood coupled with an approximate posterior (Liu et al. 2020), rendering $(\mathbf{f}|\mathbf{u},X,Z)$ deterministic during the training phase. Letting $A = K_{XZ}K_{ZZ}^{-1}$ and reparameterizing $\mathbf{u} = \boldsymbol{\mu}_q + L_q\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon}\sim N(\mathbf{0},\mathrm{I})$, we can rewrite the ELBO in closed form:

$$\begin{aligned} 2\mathcal{L} = &- 2N\log\sigma - \sigma^{-2}(\|\boldsymbol{y}\|_2^2 - 2\boldsymbol{y}^\top A\boldsymbol{\mu}_q + \|A\boldsymbol{\mu}_q\|_2^2 \\ &+ \|AL_q\mathbf{1}\|_2^2) - \log|K_{ZZ}| + 2\log|L_q| + M \\ &- \mathrm{tr}(K_{ZZ}^{-1}L_qL_q^\top) - \boldsymbol{\mu}_q^\top K_{ZZ}^{-1}\boldsymbol{\mu}_q \end{aligned} \quad (7)$$

where $\mathbf{1}$ is a column vector of ones. We can then compute the partial derivatives of $\mathcal{L}$ w.r.t. the parameters of the proposal posterior $q(\mathbf{u}|X,Z)$ (*i.e.,* $\{\boldsymbol{\mu}_q, L_q\}$), yielding:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\mu}_q} = \frac{1}{\sigma^2}(-A^\top\boldsymbol{y} + A^\top A\boldsymbol{\mu}_q) + K_{ZZ}^{-1}\boldsymbol{\mu}_q = 0 \quad (8)$$

$$\frac{\partial\mathcal{L}}{\partial L_q} = \frac{1}{\sigma^2}A^\top AL_q\mathbf{1}\mathbf{1}^\top + (L_q^{-\top} + K_{ZZ}^{-1}L_q) = 0 \quad (9)$$

Solving the above equations gives:

$$\boldsymbol{\mu}_q = \sigma^{-2}K_{ZZ}BK_{XZ}^\top\boldsymbol{y} \quad (10)$$

$$L_q(\mathrm{I} + \mathbf{1}\mathbf{1}^\top) = K_{ZZ}BK_{ZZ} \quad (11)$$

with $B = (K_{ZZ} + \sigma^{-2}K_{XZ}^\top K_{XZ})^{-1}$. To solve the triangular matrix L_q from (11), we first compute the Cholesky decomposition of $\mathrm{I}+\mathbf{1}\mathbf{1}^\top = CC^\top$ and $K_{ZZ}BK_{ZZ} = UU^T$. We then simplify both side of (11) to $L_qC = U$. L_q can then be solved by exploiting the triangular structure on both side with

$$L_{i,i-k} = \frac{U_{i,i-k} - \sum_{j=0}^{k-1} L_{i,i-j}C_{i-j,i-k}}{C_{i-k,i-k}} \quad (12)$$

where $k = 0,\cdots,i-1$, $L_{i,j}$ is a shorthand for $[L_q]_{i,j}$. We separate the model parameters into two groups, *i.e.,* parameters w.r.t. the proposal posterior $\{\boldsymbol{\mu}_q, L_q\}$ and the remaining parameters Θ, and use an EM-like algorithm to update both groups alternatively. The L-DKGPR algorithm is listed in Algorithm 1.

Prediction. Given the covariate matrix X_* for the test data, the predictive distribution is given by:

$$\begin{aligned} p(\mathbf{f}_*|X_*,X,y,Z) \simeq \mathcal{N}(&K_{X_*Z}(K_{ZZ} + \sigma^2\mathrm{I})^{-1}\boldsymbol{\mu}_q, \\ &K_{X_*X_*} - K_{X_*Z}(K_{ZZ} + \sigma^2\mathrm{I})^{-1}K_{X_*Z}^\top) \end{aligned} \quad (13)$$

Complexity. The time complexity and space complexity of both inference and prediction are $\mathcal{O}(NM^2)$ and $\mathcal{O}(NM)$ respectively, where N is the number of measured outcomes, and M the number of inducing points.

Algorithm 1: L-DKGPR

Input: Training set $S = \{X, \boldsymbol{y}\}$, latent dimension D_v, D_i, number of inducing points M, gradient-based optimizer and its related hyper-parameters (*i.e.,* learning rate, weight decay, mini-batch size), alternating frequency T.

```
Initialize the parameters Θ = {σ², Z, α^(v), α^(i), γ, φ}
while Not converged do
    Update proposal posterior q(u|X, Z) according to (10) and (12)
    t = 0
    for t < T do
        Update Θ using the input optimizer.
        t = t + 1
```

Experiments

We compare L-DKGPR to several state-of-the-art LDA and GP methods on simulated as well as real-world benchmark data. The experiments are designed to answer research questions about accuracy, scalability, and interpretability of L-DKGPR: (RQ1) How does the performance of L-DKGPR compare with the state-of-the-art methods on standard longitudinal regression tasks? (RQ2) How does the scalability of L-DKGPR compare with that of the state-of-the-art longitudinal regression models? (RQ3) Can L-DKGPR reliably recover the rich correlation structure from the data? (RQ4) How do the different components of L-DKGPR contribute to its overall performance? (RQ5) What is the advantage of solving the exact ELBO in (7) compared to solving its original form in (6) using Monte Carlo sampling (Wilson et al. 2016b)?

Data

We used one simulated data set and three real-world longitudinal data sets in our experiments:[1]

Simulated data. We construct simulated longitudinal data that exhibit *i.e.,* longitudinal correlation (LC) and multilevel correlation (MC) as follows: The outcome is generated using $\boldsymbol{y} = f(X) + \boldsymbol{\epsilon}$ where $f(X)$ is a non-linear transformation based on the observed covariate matrix X and the residual $\boldsymbol{\epsilon}\sim N(\mathbf{0},\Sigma)$. To simulate longitudinal correlation, we simply set Σ to a block diagonal matrix with non-zero entries for within-individual observations. To simulate multilevel correlation, we first split the individuals into C clusters and assign non-zero entries for the data points in the same cluster. Following (Cheng et al. 2019; Timonen et al. 2019), we simulate 40 individuals, 20 observations, and 30 covariates for each individual. We vary the number of clusters C from $[2,5]$.

Study of Women's Health Across the Nation (SWAN) (Sutton-Tyrrell et al. 2005). SWAN is a multi-site longitudinal study designed to examine the health of women during the midlife years. We consider the task of predicting the CESD

[1]Details of generation of simulated data and of pre-processing of real-world data are provided in the Appendix.

score, which is used for screening for depression. Similar to (Liang et al. 2020), we define the adjusted CESD score by $y = \text{CESD} - 15$, thus $y \geq 0$ indicates depression. The variables of interest include aspects of physical and mental health, and demographic factors such as race and income. The resulting data set has $3,300$ individuals, 137 variables and $28,405$ records.

General Social Survey (GSS) (Smith et al. 2017). The GSS data were gathered over 30 years on contemporary American society collected with the goal of understanding and explaining trends and constants in attitudes, behaviors, and attributes. In our experiment, we consider the task of predicting the self-reported general happiness of $4,510$ individuals using $1,553$ features and $59,599$ records. We follow the experimental setup in (Liang et al. 2020), with $y = 1$ indicates happy and $y = -1$ indicates the opposite.

The Alzheimer's Disease Prediction Of Longitudinal Evolution (TADPOLE) (Marinescu et al. 2018). The TADPOLE challenge involves predicting the symptoms related to Alzheimer's Disease (AD) within 1-5 years of a group of high-risk subjects. In our experiment, we focus on predicting the ADAS-Cog13 score using the demographic features and MRI measures (Hippocampus, Fusiform, WholeBrain, Entorhinal, and MidTemp). The resulting data set has $1,681$ individuals, 24 variables and $8,771$ records.

Experimental Setup

To answer RQ1, we use both simulated data and real-world data. To evaluate the regression performance, similar to (Liang et al. 2020), we compute the mean and standard deviation of R^2 between the actual and predicted outcomes of each method on each data set across 10 independent runs. We use 50%, 20%, 30% of data for training, validation, and testing respectively.

To answer RQ2, we take data from a subset consisting of 50 individuals with the largest number of observations from each real-world data. We record the run time per iteration of each method on both the 50-individual subset and full data set. Because not all baseline methods implement GPU acceleration, we compare the run times of all the methods without GPU acceleration. We report execution failure if a method fails to converge within 48 hours or generates an execution error (Liang et al. 2020).

To answer RQ3, we rely mainly on the simulated data since the actual correlation structures underlying the real-world data sets are not known. We evaluate the performance of each method by visualizing the learned correlation matrix and compare it to the ground truth correlation matrix on simulated data. Additionally, we illustrate how the correlation matrix learned by L-DKGPR can provide gain useful insights using a case study with the SWAN data. Results for case study is presented in the Appendix.

To answer RQ4, we compare the performance of L-DKGPR with L-RBF-GPR, a variant that replaces the learned deep kernel with a simple RBF kernel; and L-DKGPR-, a variant of L-DKGPR without the time-invariant effects.

To answer RQ5, we compare the regression performance and hyper-parameter choices of L-DKGPR solved using Algorithm 1 with the version of L-DKGPR solved using Monte Carlo sampling (Wilson et al. 2016b) on SWAN and GSS data sets.

Baseline Methods We compare L-DKGPR with the following baseline methods: (i) Conventional longitudinal regression models, *i.e.,* **GLMM** (Bates et al. 2015) and **GEE** (Inan and Wang 2017); (ii) State-of-the-art longitudinal regression models, *i.e.,* **LMLFM** (Liang et al. 2020) and **LGPR** (Timonen et al. 2019); (iii) State-of-the-art Gaussian Process models for general regression, *i.e.,* **KISSGP** with deep kernel (Wilson et al. 2016b) (we use the same deep structure as in our time-varying kernel) and **ODVGP** (Salimbeni et al. 2018). Implementation details[2] and hyper-parameter settings of L-DKGPR as well as the baseline approaches are provided in the Appendix.

Results

We report the results of our experiments designed to answer the research questions RQ1-RQ4.

L-DKGPR vs baseline longitudinal regression methods. The results are reported in Table 1 and Table 2 for simulated and real-world data sets respectively. In the case of simulated data, we find that KISSGP, ODVGP, GEE and GLMM fail in the presence of MC with the mean R^2 being negative (indicative of models containing variables that are not predictive of the response variable). This can be explained by the fact that GEE is designed only to handle pure LC, thus fails to account for CC or MC. While GLMM is capable of handling MC, it requires practitioners to specify the cluster structure responsible for CC prior to model fitting. However, in our experiments, the cluster structure is unknown a priori. Hence it is not surprising that GLMM performs poorly. Though both KISSGP and ODVGP are conceptually viable to handle data with complex correlation, they both experience dramatic performance drop when cluster correlation (or time-invariant effects) are presented. Moreover, we find that although LMLFM outperforms GLMM and GEE in the presence of MC, its R^2 is still quite low. This is because LMLFM accounts for only a special case of MC, namely, for CC among individuals observed at the same time points, and not all of the CC present in the data. We find that LGPR performs rather poorly on both simulated and real-world data. This might due to the fact that LGPR obtains the contributions of each variable to the kernel independently before calculating their weighted sum. Though it is possible to incorporate higher-order interactions between variables into LGPR, doing so requires estimating large numbers of interaction parameters, with its attendant challenges, especially when working with small populations. In contrast to the baseline methods, L-DKGPR consistently and significantly outperforms the baselines by a large margin. On the real-world data sets, L-DKGPR outperforms the longitudinal baselines in most of the cases.

Scalability of L-DKGPR vs. baseline methods. We see from Table 2 that most longitudinal baselines, *i.e.,* LGPR, GLMM, and GEE, fail to process real-world data sets with large numbers of covariates. Indeed, their computational complexity increases in proportion to P^3 where P is the number

[2]Data and codes used in this paper are publicly available at https://github.com/junjieliang672/L-DKGPR.

Table 1: Regression accuracy R^2 (%) comparison on simulated data with different correlation structures.

Method	LC	MC($C = 2$)	MC($C = 3$)	MC($C = 4$)	MC($C = 5$)
L-DKGPR	**86.0±0.2**	**91.3±0.2**	**99.6±0.2**	**99.8±0.2**	**99.8±0.2**
KISSGP	85.9±1.7	-43.4±33.3	-55.5±7.1	-58.2±14.4	-57.2±17.9
ODVGP	82.3±5.2	-1.6±16.9	-14.7±6.5	-13.5±8.4	-6.1±4.4
LGPR	-37.1±19.1	-123.6±162.0	-26.3±43.2	-9.1±14.8	-0.1±5.9
LMLFM	54.7±15.1	-138.3±121.9	-48.3±123.6	22.6±49.0	36.2±41.1
GLMM	5.3±27.9	-656.3±719.8	-801.4±507.4	-684.1±491.3	-528.7±313.5
GEE	59.0±24.5	-636.1±606.0	-703.6±465.8	-665.6±554.3	-516.5±457.5

Table 2: Regression accuracy R^2 (%) on real-world data sets. We use 'N/A' to denote execution error.

Data sets	N	I	P	L-DKGPR	KISSGP	ODVGP	LGPR	LMLFM	GLMM	GEE
TADPOLE	595	50	24	44.0±5.6	1.2±10.1	9.0±14.1	-261.1±9.0	8.7±5.1	**50.8±5.5**	-11.4±4.8
SWAN	550	50	137	**46.8±4.9**	42.4±4.6	29.0±3.1	-16.6±12.7	38.6±4.2	40.1±7.7	46.4±8.0
GSS	1,500	50	1,553	**19.1±3.7**	12.5±6.3	-7.6±3.3	N/A	15.3±1.4	N/A	-4.6±3.5
TADPOLE	8,771	1,681	24	**64.9±1.4**	0.6±3.9	21.1±1.0	N/A	10.4±0.6	61.9±1.9	17.6±0.7
SWAN	28,405	3,300	137	**52.5±0.4**	20.5±7.6	24.9±21.8	N/A	48.6±2.0	N/A	N/A
GSS	59,599	4,510	1,553	**56.9±0.1**	53.1±0.9	15.4±27.0	N/A	54.8±2.2	N/A	N/A

of covariates. In contrast, L-DKGPR, LMLFM and state-of-the-art GP baselines (KISSGP and ODVGP) scale gracefully with increasing number of data points and covariates. For CPU run time analysis, please refer to our Appendix.

Recovery of Correlation Structure. The outcome correlations estimated by all GP methods on the simulated data are shown in Figure 2. We see that KISSGP and ODVGP are incapable of recovering any correlation structure from the data. LGPR seems to be slightly better than KISSGP and ODVGP when MC is presented. However, we see that only one known cluster is correctly recovered when $C > 2$. This suggests that these methods fail to recover accurate correlation structures, which is consistent with their poor performance in terms of R^2. In contrast, L-DKGPR is able to recover most of the correlation structure present in the data. It is worth noting that recovering correlation structure is a challenging task and although L-DKGPR is the best performing model, the learned correlation structure is still far from perfect. A possible explanation is that without further prior constraints on the kernel structure, the kernel search space is very large. Since L-DKGPR works in an MLE framework, it searches for a kernel to improve the likelihood. When optimal solution is surrounded by infinitely many local maxima, each with simpler kernel structure but sufficiently high likelihood, it is not surprising that L-DKGPR gets stuck in one of such local maxima since the kernel initialization of L-DKGPR is uninformative.

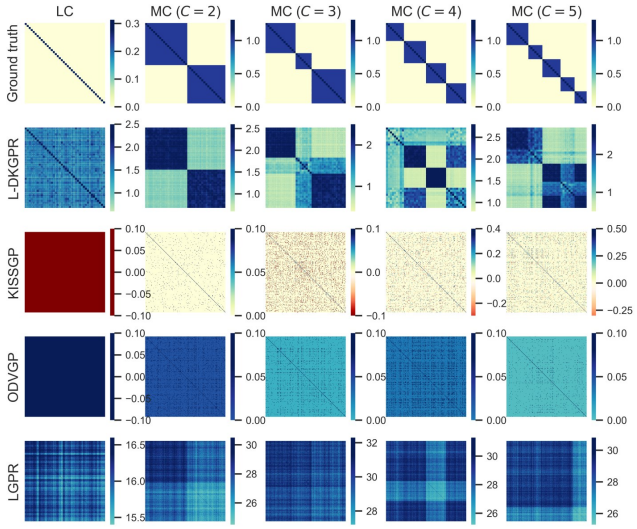

Figure 2: Outcome correlation estimated by all GP methods on simulated data.

Ablation study. Regression accuracy comparison on complete real-world data sets is shown in Table 3. **Role of time-invariant component**: We see a dramatic drop in regression performance when time-invariant effects are not modeled (L-DKGPR-v) as compared to when they are (L-DKGPR). This result underscores the importance of modeling the time-independent components of LC and CC for accurate modeling of longitudinal data. This task is simplified by the decomposition of the correlation structure into the time-varying and time-invariant components. The time-invariant component is analogous to estimating the mean correlation whereas the time-varying component contributes to the residual. Hence, the decomposition of the correlation structure into time-varying and time-invariant components should help reduce the variance of the correlation estimates. **Role of time-varying component**: We observe significant performance drop when time-varying effects are not modeled (L-DKGPR-i) as compared to L-DKGPR. This is reasonable

Table 3: Effect on the regression accuracy R^2 (%) of different components of L-DKGPR

Data sets	L-DKGPR	L-DKGPR-v	L-DKGPR-i	L-RBF-GPR
TADPOLE	**64.9±1.4**	13.2±1.1	56.3±1.3	55.5±2.4
SWAN	**52.5±0.4**	29.0±3.2	16.7± 2.4	5.4±1.6
GSS	**56.9±0.1**	56.2±0.1	-0.2±0.2	-14.1±0.4

Table 4: Effect of solving L-DKGPR using Algorithm 1 vs. Monte Carlo sampling.

Data sets	Solver	M	Iterations	R^2 (%)
SWAN	Alg. 1	10	300	**52.5±0.4**
	Sampling	10	300	3.1±0.2
	Sampling	128	3,000	51.4±0.4
GSS	Alg. 1	10	300	**56.9±0.1**
	Sampling	10	300	4.5±0.1
	Sampling	128	3,000	55.6±0.1

because without the time-varying kernel, the model gives the same outcome prediction for an individual at all time. This is unrealistic for longitudinal data. **Role of deep kernel**: L-DKGPR consistently outperforms L-RBF-GPR (which uses RBF kernel instead of the deep kernel used by L-DKGPR), with the performance gap between between the two increasing with increase in the number of covariates. This is perhaps explained by the pitfalls of Euclidean distance as a measure of similarity between data points in a high dimensional data space (Aggarwal, Hinneburg, and Keim 2001) (and hence kernels such as the RBF kernel which rely on Euclidian distance in the data space), and the apparent ability of the learned deep kernel to perform such similarity computations in a low-dimensional latent space where the computed similarities are far more reliable.

Effect of solving the exact ELBO with Algorithm 1. Table 4 presents the results in comparing L-DKGPR solved using Algorithm 1 with a version of L-DKGPR solved using the vanilla Monte Carlo sampling (Wilson et al. 2016b). We find that under the same hyper-parameter setting, our solver outperforms the sampling solver by a large margin. To ensure similar regression performance, we have to modify the hyper-parameters for the sampling solver by increasing the number of inducing points M to 128 and using about 10 times more training iterations. The result indicates that coping with the variance of the noisy ELBO approximation increases the number of parameters and hence the number of iterations needed.

Effect of the number of inducing points M. Inducing points provide a trade-off between approximation accuracy and efficiency in sparse GP methods. In this experiment, we vary the number of inducing points M from 5 to 100 on simulated data and record the R^2 as shown in Figure 3. We find that when the number of inducing points reaches a certain threshold, *i.e.,* 10 in *all* simulated settings, regression performance is rather stable, an observation that is supported by our

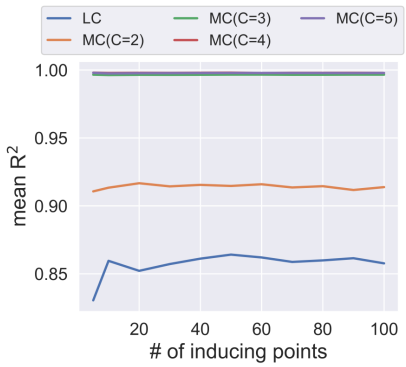

Figure 3: Regression performance with different numbers of inducing points on simulated data.

experiments with real-world data as well (results omitted). A theoretical study (Burt, Rasmussen, and Van Der Wilk 2019) points out that when input data are normally distributed and inducing points are drawn from a k-deterministic point process with an SE-ARD kernel, then $M = \mathcal{O}(\log^P N)$. In our case, since the inducing points lie in the latent space, the number of inducing points suffice to our simulated data should be as large as $M = [2\log(40)]^{10}$. In contrast, we empirically show that $M \approx \log N$ is sufficient to get consistent and appealing results. We conjecture that this is because instead of drawing the inducing points from a k-DPP process from the input data, we optimize representation of the inducing points jointly with the other model parameters, thus delivering more effective inducing points that summarize the variance of the input data. Proving or disproving this conjecture would require a deeper theoretical analysis of L-DKGPR.

Conclusion

We have presented L-DKGPR, a novel longitudinal deep kernel Gaussian process regression model that overcomes some of the key limitations of existing state-of-the-art GP regression methods for predictive modeling from longitudinal data. L-DKGPR fully automates the discovery of complex multilevel correlations from longitudinal data. It incorporates a deep kernel learning method that combines the expressive power of deep neural networks with the flexibility of non-parametric kernel methods, to capture the complex multilevel correlation structure from longitudinal data. L-DKGPR uses a novel additive kernel that simultaneously models both time-varying and the time-invariant effects. We have shown how L-DKGPR can be efficiently trained using latent space inducing points and the stochastic variational method. We report results of extensive experiments using both simulated and real-world benchmark longitudinal data sets that demonstrate the superior predictive accuracy as well as scalability of L-DKGPR over the state-of-the-art LDA and GP methods. A case study with a real-world data set illustrates the potential of L-DKGPR as a source of useful insights from complex longitudinal data.

Acknowledgements: This work was funded in part by the NIH NCATS grant UL1 TR002014 and by NSF grants 2041759, 1636795, the Edward Frymoyer Endowed Professorship at Pennsylvania State University and the Sudha Murty Distinguished Visiting Chair in Neurocomputing and Data Science funded by the Pratiksha Trust at the Indian Institute of Science (both held by Vasant Honavar).

References

Aggarwal, C. C.; Hinneburg, A.; and Keim, D. A. 2001. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, 420–434. Springer.

Bates, D.; Mächler, M.; Bolker, B.; and Walker, S. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software, Articles* 67(1): 1–48. ISSN 1548-7660.

Burt, D. R.; Rasmussen, C. E.; and Van Der Wilk, M. 2019. Rates of convergence for sparse variational Gaussian process regression. *arXiv preprint arXiv:1903.03571* .

Cheng, L.; Ramchandran, S.; Vatanen, T.; Lietzén, N.; Lahesmaa, R.; Vehtari, A.; and Lähdesmäki, H. 2019. An additive Gaussian process regression model for interpretable non-parametric analysis of longitudinal data. *Nature communications* 10(1): 1798.

Duvenaud, D. K.; Nickisch, H.; and Rasmussen, C. 2011. Additive gaussian processes. *Advances in neural information processing systems* 24: 226–234.

Fitzmaurice, G. M.; Laird, N. M.; and Ware, J. H. 2012. *Applied longitudinal analysis*, volume 998. John Wiley & Sons.

Gibbons, R. D.; and Hedeker, D. 1997. Random effects probit and logistic regression models for three-level data. *Biometrics* 1527–1537.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.

Hedeker, D.; and Gibbons, R. D. 2006. *Longitudinal data analysis*, volume 451. John Wiley & Sons.

Hsieh, T.-Y.; Wang, S.; Sun, Y.; and Honavar, V. 2021. Explainable Multivariate Time Series Classification: A Deep Neural Network Which Learns To Attend To Important Variables As Well As Informative Time Intervals .

Inan, G.; and Wang, L. 2017. PGEE: An R Package for Analysis of Longitudinal Data with High-Dimensional Covariates. *R Journal* 9(1): 393–402.

Liang, J.; Xu, D.; Sun, Y.; and Honavar, V. 2020. LMLFM: Longitudinal Multi-Level Factorization Machines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34.

Liang, K.-Y.; and Zeger, S. L. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73(1): 13–22.

Liu, H.; Ong, Y.-S.; Shen, X.; and Cai, J. 2020. When Gaussian process meets big data: A review of scalable GPs. *IEEE Transactions on Neural Networks and Learning Systems* .

Marinescu, R. V.; Oxtoby, N. P.; Young, A. L.; Bron, E. E.; Toga, A. W.; Weiner, M. W.; Barkhof, F.; Fox, N. C.; Klein, S.; Alexander, D. C.; et al. 2018. TADPOLE Challenge: Prediction of Longitudinal Evolution in Alzheimer's Disease. *arXiv preprint arXiv:1805.03909* .

McCulloch, C. E. 1997. Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American statistical Association* 92(437): 162–170.

Quintana, F. A.; Johnson, W. O.; Waetjen, L. E.; and B. Gold, E. 2016. Bayesian nonparametric longitudinal data analysis. *Journal of the American Statistical Association* 111(515): 1168–1181.

Rasmussen, C. E. 2003. Gaussian processes in machine learning. In *Summer School on Machine Learning*, 63–71. Springer.

Salimbeni, H.; Cheng, C.-A.; Boots, B.; and Deisenroth, M. 2018. Orthogonally decoupled variational gaussian processes. In *Advances in neural information processing systems*, 8711–8720.

Seeger, M.; Williams, C. K.; and Lawrence, N. D. 2003. Fast Forward Selection to Speed Up Sparse Gaussian Process Regression. In *Workshop on AI and Statistics*.

Smith, T.; Marsden, P.; Hout, M.; and Kim, J. 2017. General Social Surveys, 1972–2014 [machine-readable data file]/Principal Investigator. *Sponsored by national science foundation. Chicago: National Opinion Research Center at the University of Chicago [producer and distributor]* .

Sutton-Tyrrell, K.; Wildman, R. P.; Matthews, K. A.; Chae, C.; Lasley, B. L.; Brockwell, S.; Pasternak, R. C.; Lloyd-Jones, D.; Sowers, M. F.; Torréns, J. I.; et al. 2005. Sex hormone–binding globulin and the free androgen index are related to cardiovascular risk factors in multiethnic premenopausal and perimenopausal women enrolled in the Study of Women Across the Nation (SWAN). *Circulation* 111(10): 1242–1249.

Tang, X.; Yao, H.; Sun, Y.; Aggarwal, C. C.; Mitra, P.; and Wang, S. 2020. Joint Modeling of Local and Global Temporal Dynamics for Multivariate Time Series Forecasting with Missing Values. In *AAAI*, 5956–5963.

Timonen, J.; Mannerström, H.; Vehtari, A.; and Lähdesmäki, H. 2019. An interpretable probabilistic machine learning method for heterogeneous longitudinal studies. *arXiv preprint arXiv:1912.03549* .

Verbeke, G.; Fieuws, S.; Molenberghs, G.; and Davidian, M. 2014. The analysis of multivariate longitudinal data: A review. *Statistical methods in medical research* 23(1): 42–59.

Wang, K.; Pleiss, G.; Gardner, J.; Tyree, S.; Weinberger, K. Q.; and Wilson, A. G. 2019. Exact Gaussian processes on a million data points. In *Advances in Neural Information Processing Systems*, 14648–14659.

Wang, M. 2014. Generalized estimating equations in longitudinal data analysis: a review and recent developments. *Advances in Statistics* 2014.

Williams, C. K.; and Rasmussen, C. E. 2006. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Wilson, A. G.; Hu, Z.; Salakhutdinov, R.; and Xing, E. P. 2016a. Deep kernel learning. In *Artificial Intelligence and Statistics*, 370–378.

Wilson, A. G.; Hu, Z.; Salakhutdinov, R. R.; and Xing, E. P. 2016b. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, 2586–2594.

Xiong, Y.; Kim, H. J.; and Singh, V. 2019. Mixed Effects Neural Networks (MeNets) With Applications to Gaze Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7743–7752.