# Nonparametric Gaussian Process Estimation R Package with an Application to Functional Linear Models

**Robert Pehlman**
Department of Statistics
North Carolina State University
Raleigh, NC 27695
rpehlma@ncsu.edu

## Abstract

The aim of this project was to create an R package to do nonparametric gaussian process estimation. The estimation procedure takes as input sparse, irregular longitudinal data, which are assumed to be subject to measurement error, and outputs an estimate for the covariance kernel of the process. An application of Gaussian process estimation to functional linear models is explored. We examine computational complexity of the algorithm under differing circumstances.

## 1 Introduction

The computational problem is to compute the covariance kernel of a stochastic process. This is difficult because the true covariance kernel may be infinite dimensional, and the observed data used to estimate the process may be longitudinal data are sparse, irregularly spaced, noisy data, which may vary in number among observations. In practice it is common to assume a completely parametric model. If we want a nonparametric covariance model, we can approximate the kernel using a finite basis. The best approximation to a kernel given a fixed number of p basis functions is the sum of the first eigenfunction / eigenvalue pairs of the Karhunen-Loeve decomposition. The KarhunenLove theorem provides representation of a stochastic process as an infinite linear combination of orthogonal functions,

$$Z(t) = \sum_{i=1}^{\infty} \xi_i \phi_i(t),$$

which has the following properties:

- The functions, $\{\phi_i(t)\}_{i=1}^{\infty}$, are deterministic and orthogonal ($\int_0^1 \phi_i(t)\phi_j(t)dt = \delta_{i,j}$).
- The random variables $\xi_i$ are independent $N(0, \lambda_i)$, with $\lambda_1 > \lambda_2 > ....$
- The finite truncation of $Z(t)$ resulting from taking the first p (function, RV) pairs, $\hat{Z} = \sum_{i=1}^{p} \xi_i \phi_i(t)$ minimizes the mean squared error between the approximation and $Z(t)$
- The pairs $(\phi_i(t), \lambda_i)$ are the respective corresponding eigenfunctions and eigenvalues of the covariance kernel, $K(s,t)$.

We could find the correct kernel approximation using maximum likelihood. The difficulty of this optimization comes from the fact that the basis functions we seek must be orthogonal. We can ensure this constraint by considering orthogonal transformations of a basis set of orthogonal functions. Consequently, we have a non-convex parameter space, so standard methods cannot be applied.

We solve this issue by using projected steepest descent, so that our maximum likelihood iterates never leave the function space. We compare this method to two solvers for manifold problems that are in the R package manifoldOptim.

## 2 Methodology

The data comes in the form of sparse, noisy, irregularly spaced longitudinal measurements which may be unequal in number. An example curve and the data we observe are shown in Figure 1.

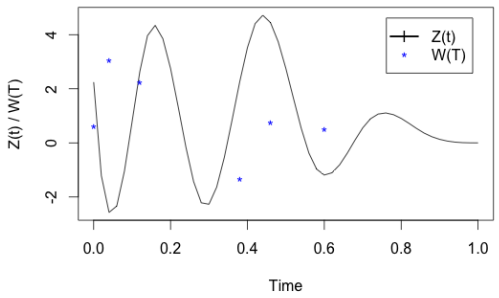


Figure 1: Example Longitudinal Data.

The longitudinal measurements we observe for subject i are a $M_i$-variate gaussian random vector, which we assume to be mean zero and whose covariance matrix is determinedby the measurement times $T_i$. Then the product of these gaussians forms a likelihood function, and we maximize the log-likelihood. The procedure is an iterative algorithm. At each step we calculate the gradient $\nabla \ell(\theta, W(T))$, call it $\nabla f(\theta)$. Then the sequence of iterates is defined by

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} + \alpha \nabla f(\theta^{(k)})$$

where $\alpha$ is found by by testing a discrete set of candidate $\alpha$'s and choosing the one which maximizes $f(\theta^{(k+1)})$. In our example the algorithm runs for a fixed number of 300 iterates, but parameters exist in the r package to stop the algorithm based on achieving some given tolerance.

We apply these estimates to the problem of estimating a functional linear model. Analogous to a linear model, a simple functional linear model can be written as

$$E[Y|Z(t)] = \int_0^1 \beta(t) Z(t) dt,$$

where Y is a scalar outcome, $\beta(t)$ is a deterministic coefficient function, and $Z(t)$ is a stochastic process, in our example a gaussian process. For this example we will consider functions of a single variable which are continuous, and bounded on the interval [0,1].

In a typical linear model problem, it is assumed that the $\beta$ is unknown, but the covariates are known. In our problem, we assume that $\beta(t)$ is unknown and that $Z(t)$ must be estimated based on the observed longitudinal data. In a "Functional feature construction for individualized treatment regimes"[3], the authors propose a method to approximate $Z(t)$ using the Karhunen-Loeve representation. Using the estimated covariance kernel, we find the conditional expectations, given W(T), of the p largest-variance random variables in the Karhuhen-Loeve decomposition of the stochastic process. As the number of orthogonal basis functions grows the coefficient function, $\beta(t)$, can be approximated arbitrarily well by linear combinations of functions from $\{\phi_i(t)\}_{i=1}^{L}$, the set of basis functions. This gives rise to the following approximation:

$$E[Y|Z(t)] = \int_0^1 \beta(t) Z(t) dt = \int_0^1 \left( \sum_{i=1}^{\infty} \beta_i \phi_i(t) \right) \left( \sum_{j=1}^{\infty} \xi_j \phi_j(t) \right) dt$$

Table 1: Average times as n increases, low density

| n | RWRBFGS | RTRSR1 | PSD |
|---|---|---|---|
| 100 | 13.84 | 8.18 | 9.75 |
| 200 | 35.04 | 18.91 | 13.79 |
| 400 | 108.13 | 50.74 | 27.555 |

Table 2: Average times as n increases, high density

| n | RWRBFGS | RTRSR1 | PSD |
|---|---|---|---|
| 100 | 26.31 | 32.55 | 73.2 |
| 200 | 68.97 | 85.88 | 146.86 |
| 400 | 147.42 | 179.3 | 428.99 |

$$\approx \int_0^1 \left(\sum_{i=1}^{L} \beta_i \phi_i(t)\right)\left(\sum_{j=1}^{L} \xi_j \phi_j(t)\right) = \sum_{i=1}^{L} \beta_i \xi_i$$

due to orthogonality. Taking conditional expectations WRT. W(T) on both sides, $E[Y|W(T)] = E\{E[Y|Z(t)]|W(T)\} \approx \sum_{i=1}^{L} \beta_i E[\xi_i|W(T)]$, an ordinary linear model.

The results of this procedure, along with an outcome, Y, of interest which depends on the process $Z(t)$, are used to fit a linear model and estimate the coefficient function which weights the stochastic process.

## 3 Analysis and Results

The maximum likelihood problem involves optimization over the space of orthonormal $(L \times p)$ matrices, which is referred to as the Stiefel Manifold. The set of orthonormal matrices is not convex, so special machinery such as projected steepest descent and Riemannian BFGS are used to maximize the likelihood.

We compare 3 different optimization algorithms for finding the maximum likelihood estimates. Projected Steepest Descent (PSD), Riemannian BFGS (RWRBFGS) [2], and a Riemannian trust-region method with a symmetric rank-one update (RTRSR1) [1].

We will consider as a standard problem the case when n = 100, the number of selected eigenfunctions in the approximation p = 5, the number of orthogonal B-Spline basis functions L = 10, the average number of observations per subject, E(d) is 10 for low density and 35 for high density, and $[0, 1]$ is discretized into 51 time points.

Out of 20 trials on the standard setting, PSD method failed to converge most frequently, at 6 in 20 trials. RWRBFGS failed to converge in 4 of 20 trials, RTRSR1 failed to converge in 2 in 20 trials. PSD failed to converge in a similar fashion every time, by incorrectly estimating the parameters for the eigenfunctions and eigenvalues associated with the smallest eigenvalues. Typically where RTRSR1 failed to converge, RWRBFGS succeeded in converging. This suggests a good practice would be to use RTRSR1 and RWRBFGS and take the result with the better likelihood value.

As seen in Table 1, for all 3 optimization algorithms, the change in runtime as a function of the number of subjects is roughly linear (actually a little faster than linear). PSD is more constant because it always ran to the max number of iterations and the others converged based on a tolerance first.

At a higher data density the complexity should be cubic in $E(d)$, but the cost of inverting s $E(d) \times E(d)$ matrix is dominated by start up costs, so that is not seen here. See Table 2.

# 4 Discussion

- RTRSR1 and RWRBFGS perform better than PSD in terms of speed and accuracy.
- The theoretical problem complexity should be $O(nd^3)$ per iteration because a $d \times d$ matrix needs to be cholesky factorized n times, but in our numerical examples this operation was dwarfed by the creation of numerous $pxp$ matrices, so p was the primary driver of time increases.
- Speed is quick enough for medium sized problem to run on a consumer grade laptop in a few minutes. The longitudinal observations per subject can be downsampled to increase speed without impacting the convergence properties as long as they form a dense set in [0,1].

## 4.1 Forthcoming Research

Future work on this project will extend the estimation procedure to cover the case when a treatment is received during the course of the process and the estimated parameters of the process depend on the treatment given as well as the time of treatment. This allows for a more interesting Q-learning application, because a decision about the best treatment can be based on a process and optimized while the longitudinal data is still being collected, instead of needing to be fully observed to be summarized and used as input in the Q-function. This approach requires additional assumptions about the form of the covariance kernel, because the portion of the process after treatment is administered must be dependent on the portion before treatment in order to gain any predictive power from the estimation procedure.

# References


[1] Wen Huang, P. A. Absil, and K. A. Gallivan. A riemannian symmetric rank-one trust-region method. *Math. Program.*, 150(2):179–216, May 2015.

[2] Wen Huang, K. A. Gallivan, and P.-A. Absil. A broyden class of quasi-newton methods for riemannian optimization. *SIAM Journal on Optimization*, 25(3):1660–1685, 2015.

[3] Eric B. Laber and Ana-Maria Staicu. Functional feature construction for individualized treatment regimes. *JASA (In Press)*.