UNIVERSITY OF CALIFORNIA
RIVERSIDE

Simultaneous Confidence Bands for Monte Carlo Simulations

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Statistics

by

Jinhui Yang

June 2021

Dissertation Committee:

Dr. James Flegal, Chairperson
Dr. Esra Kurum
Dr. Nanpeng Yu

The Dissertation of Jinhui Yang is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am thrilled to take this journey from the first day as a PhD student until the last day with this dissertation. There are so many people I would want to thank.

Great thanks to my advisor Dr. James Flegal for all the support. Dr. Flegal provided great research opportunities with helpful advice and creative ideas. I am grateful to Dr. Flegal for all the help throughout these years.

I would want to thank Dr. Nanpeng Yu and Dr. Esra Kurum as my dissertation committee members. Special thanks to Dr. Yu who supervised me for my electric engineering project with the encourages and guidance.

Thanks to Dr. Weixin Yao for his detailed instructions in my course work as my graduate advisor.

I am also grateful to Dr. Gloria Gonzalez-Rivera, Dr. Subir Ghosh and Dr. Gregory Palardy as my oral exam committee members. They supported me through my PhD period in various ways.

I want to thank UCR Statistics Department and Graduate Division for providing the Dean's Distinguished Fellowship, Dissertation Fellowship, and Teaching Assistant opportunities together with the coursework and activities which made my PhD life educating and colorful.

Last but not least, I am grateful to my parents for their support and encouragement.

To my loved ones for all the support.

ABSTRACT OF THE DISSERTATION

Simultaneous Confidence Bands for Monte Carlo Simulations

by

Jinhui Yang

Doctor of Philosophy, Graduate Program in Applied Statistics
University of California, Riverside, June 2021
Dr. James Flegal, Chairperson

Markov Chain Monte Carlo (MCMC) methods are widely used and preferred when the sampling distribution is intractable. In estimation problems with Monte Carlo samples, it is critical to quantify uncertainty of estimators on some intervals since the true function could not be obtained in most cases. Traditional pointwise confidence intervals could provide certain coverage probability in a single point but fail to provide simultaneous coverage for the whole function without a multiplicity correction. The Bonferroni method corrects for multiplicity, but these conservative intervals do not achieve the desired nominal level. This dissertation focuses on providing and quantifying the uncertainty of estimators in the form of a confidence band (CB) to increase the reliability of the resulting inferences.

We begin with MCMC basics and point estimation methods. Then we provide estimators for densities and general functions separately. We discuss the covariance matrix and Central Limit Theorem as preliminary settings. Afterwards, we review pointwise and Bonferroni methods to construct CBs. We propose three methods in calculating simultaneous CBs with theories and algorithms which are followed by examples to compare the

coverage probabilities and the band widths. To provide more intuitive results, we compared the bands with three simulation examples: AR(1) model, mixed normal distribution, and a general function case. Then we used four real data examples: Michigan survey example, Telescope data example, time varying model, and a Bayesian reliability model to explain our proposed simultaneous bands.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In statistics, estimation problems are important. Functional estimation has been an endless topic for both frequentists and Bayesians where statistical methods for density and functional estimation are continuously developed. In these problems, it is critical that some measure of uncertainty is included so an independent reader can judge the quality of estimation.

It is critical to clarify the definition of "uncertainty" in estimation problems. We suppose f is an arbitrary function with domain $\mathbb{R}$ and f can be any function of our interest like a stock price curve varying with time. Estimation of $f(x)$ occurs at m points on a compact set $D = [a, b]$ with $\overrightarrow{w} = (w_1, \ldots, w_m)^T$ and $a = w_1 < w_2 < \cdots < w_m = b$. We want estimators $\hat{f}(w_i)$ for $f(w_i)$ with $i = 1, \ldots, m$, which is a typical estimation problem.

My focus is on the critical problem of how confident we are in estimators approximating the real function. A confidence interval is a range of plausible values for the real function at a point. In our setting, the lower bound $L(x)$ for $f(x)$ is of form $\hat{f}(x) - ME(x)$

and the upper bound $U(x)$ for $f(x)$ is of form $\hat{f}(x) + ME(x)$ for arbitrary x where $\hat{f}(x)$ is the estimator and $ME(x)$ is the margin of error that measures the difference between the real function and our estimator. The probability of the real function falling into the lower and upper bounds range is called the confidence level $1 - \alpha$

$$P\Big(L(x) \leq f(x) \leq U(x)\Big) = 1 - \alpha.$$

Pointwise confidence intervals create lower and upper bounds (L_i, U_i) for $i = 1, \ldots, m$ satisfying

$$P\Big(L_i \leq f(w_i) \leq U_i\Big) = 1 - \alpha \text{ for each i=1,...,m.}$$

Pointwise intervals have the coverage ability in the function value for each point. Such pointwise intervals fail to provide the bounds that could cover all the points "simultaneously". One approach is the Bonferroni correction with the application of the Bonferroni inequality. The Bonferroni confidence intervals satisfy

$$P\Big(L_i \leq f(w_i) \leq U_i\Big) = 1 - \frac{\alpha}{m} \text{ for each i=1,...,m.}$$

Overall the intervals yield

$$P\Big(L_i \leq f(w_i) \leq U_i \text{ for all i=1,...,m}\Big) \geq 1 - \alpha.$$

While the Bonferroni interval is a way to build the intervals for the function with the confidence level on a whole scale, it is conservative to use resulting in wide bounds. These result from the high correlations between adjacent points in the grid. Both pointwise and

Bonferroni intervals have been applied in cases with independent samples, but there are few discussions in building the simultaneous confidence intervals in correlated samples (Dunn, 1958).

Our goal is to find the lower and upper bounds that satisfy

$$P\Big(L_i \leq f(w_i) \leq U_i \text{ for all i=1,...,m}\Big) \approx 1-\alpha,$$

for correlated samples which are generated through correlated sampling from a target distribution. A popular method when a target distribution is only known up to a normalizing constant is Markov chain Monte Carlo (MCMC). In short, MCMC simulations perform estimation by constructing Markov chain with invariant distribution equal to the target distribution. Then realized draws from the chain are used for univariate and multivariate estimation by averaging conditional expectations or Rao-Blackwellization, importance sampling, marginal density estimation, and functional estimation (Andrieu et al., 2003).

Marginal density estimation is a starting topic. There are several methods to estimate a marginal density with Markov Chain strong law of large numbers (SLLN). The true density could be approximated by a histogram (Flegal and Jones, 2010a). When the full conditionals are available, we consider the Rao-Blackwellized estimation (Wei and Tanner, 1990).

More generally, we consider kernel density estimation (Parzen, 1962). Kernel density estimation provides an estimator for the density $f(x)$ with the help of well-defined kernels. One consideration about the kernel method also benefits from a bias correction procedure (Cheng and Chen, 2019). When the second derivative $f''(x)$ is not negligible in

the bias term, a proper estimation of $f''(x)$ is needed. A naive way is to estimate $f''(x)$ just from the kernel estimators which will still lead to some bias in the recurrent estimation. Another way to correct the bias is to use some bias-reducing kernels to avoid estimating $f''(x)$ (Calonico et al., 2018).

We propose methods to evaluate the uncertainty of estimation. The usually biased way to evaluate a density estimator is by building confidence intervals at a grid of points in the compact set of interest. Such pointwise confidence intervals only give information about how close the density estimator is to the true density at each single point. In order to achieve better coverage probability on the whole scale, we propose methodology to construct simultaneous confidence bands for the functions.

The problem then comes to find an estimator of the variance-covariance matrix Σ, which measures the variability among grid points. The batch mean estimator could be obtained from splitting the whole interval into several batches (Chen and Seila, 1987). With α as the significant level, a $100(1-\alpha)\%$ confidence region for θ could be given then. A lugsail batch mean estimator was suggested to provide a new way in estimating Σ (Vats and Flegal, 2018). Another improvement for the procedure is using Hotelling's T-squared distribution to replace the standard normal distribution in the central limit theorem (CLT). With certain simulations, the comparisons among these improvements are given with discussions. Evaluating uncertainty in MCMC crucially depends on the existence of a Markov chain CLT. Estimating the asymptotic variance from such a CLT requires specilized techniques that we also review.

The primary focus here is on marginal density and functional estimation, where the aim is to develop simultaneous confidence bands. That is, finding lower and upper bounds that contain an entire function (on a compact set) with some prespecified level of confidence. In this document, a procedure is suggested to evaluate the whole scale behavior of estimators by constructing confidence bands. These confidence bands are built via a parametric bootstrap procedure in conjunction with the Markov chain CLT. Estimation of the asymptotic variance covariance matrix is critical, which is accomplished via batching methods.

The pointwise interval is the commonly used method in estimating the uncertainties for the functions to achieve the desired confidence level at each point. In building the confidence bands, the classical Bonferroni corrected intervals were developed (Dunn, 1958). While the Bonferroni methods could achieve the confidence level for all the points for the independent samples, its drawbacks lie in the wide bands and non-adjustment for correlated data. In MCMC, we also need to consider the correlations among the points to be estimated.

There are increasing attentions in building confidence bands for functions to achieve the overall coverage probabilities. A method in building simultaneous confidence bands for the mean of functional data was stated in Degras (2017). Uniform confidence bands (same width for each point) were discussed in Cheng and Chen (2019). The different resampling methods to build the simultaneous regions were reviewed in Montiel Olea and Plagborg-Møller (2019). Methods of building confidence bands for densities were discussed in Chen (2017). These provide a well-developed theory for our approach using similar techniques.

We propose a bias-correction method to the kernel density approach to estimate the density and a bias-correction local linear regression method to estimate the function with the batch mean variance-covariance matrix. The traditional bias-correction method lacks the extension to the variance-covariance matrix estimation. With various applications led by kernel density estimation and function estimation, there are few discussions around how the bias-corrected kernels could contribute to evaluate the uncertainty of estimation where our method fills the gap. With solid theoretical support and simulation studies, we showed that the bias-corrected estimations improve empirical coverage probabilities.

When it comes to functional estimation and confidence bands building, a suggestion was using a local linear regression in the functional estimation (Hastie et al., 2009). On one side, it could predict the functional value at the point where there is no data available. On the other side, the bias still needs to be corrected. Though this is not stated, the bias still affects the estimation accuracy. From the foundations of local regression, we propose a method in correcting the second derivative bias. With certain degree of polynomial and bandwidth choice, a proper estimator for the function is given with the bias corrected. Along with the methodology in estimating the covariance matrix, our method could build the simultaneous confidence bands with reasonable coverage probabilities in the simulation study.

We propose methodology to correct for multiplicity with quantile-based simulations for the upper bound and lower bound in each point. The confidence bands are built through a parametric bootstrapping procedure with a quantile-related method applied. Our method achieves good overall coverage probability by comparing with pointwise, Bonferroni

and other multiplicity correction methods. The theoretical support for our methodology is from a tube formula in Wasserman (2006), see also Montiel Olea and Plagborg-Møller (2019). The quantile-based bootstrap band has a similar performance in building confidence bands as ours. With a simulation study, we have shown that the approaches are identical within certain range. Another way to build simultaneous confidence regions is to find hyperrectangular bands between pointwise intervals and Bonferroni intervals (Robertson et al., 2020). We also include this method into the comparisons.

Chapter 2

Point Estimation

2.1 Markov Chain Monte Carlo

To fully understand the methodology of density and functional estimation, we need to build some knowledge of Markov Chain Monte Carlo (MCMC) and its corresponding concepts.

In statistics, the contributions of Markovian methods can not be ignored. In time series, models like autoregressive-moving average model are highly dependent on the properties of Markov chains. In applied fields like queuing analysis, storage system and social science, Markovian models could provide deep insights into the business. In specific models like linear state space model, the Markovian methods are fundamental. In our methodology in building simultaneous confidence bands for densities and functions, Markovian methods are critical and never too deep to learn. Whenever a current value depends on the past value, the Markovian methods could find their opportunities.

A Markov chain is a dependent sequence of random variables $X_1, X_2, \ldots$ or random vectors $X_1, X_2, \ldots$ that have the property that any future X_{t+1} is independent of the past $X's$ given the present X_t for any t:

$$P(X_{t+1} = x_{t+1} | X_1 = x_1, X_2 = x_2, \ldots, X_t = x_t) = P(X_{t+1} = x_{t+1} | X_t = x_t),$$

which means that the conditional distribution of X_{t+1} given $X_1, \ldots, X_t$ depends only on X_t.

If the conditional distribution of X_{t+1} given X_t is same for all t, we say such Markov chain has stationary transition probabilities. In MCMC, every Markov chain has this property except for adaptive MCMC (Andrieu et al., 2003). Therefore the joint distribution of $X_1, X_2, \ldots, X_t$ is only determined by the X_1 and the transition probabilities which are the marginal distribution of X_1 and the conditional distribution of X_{t+1} given X_t.

Stability is a vital and basic concept in a Markov chain. In a Markov chain, a scalar functional is a time series, but may not be a Markov chain. We call a Markov chain stationary if X_1, the initial distribution is stationary. This is different from having stationary transition probabilities. Strictly speaking, all the chains in MCMC are not exactly stationary but they have stationary transition probabilities. If a Markov chain needs to be exactly stationary, the chain must be started with simulation from the equilibrium (invariant or stationary) distribution.

In a Markovian process, an important property is its transition ability, which describes the motion of a Markov chain changing from one state to another. If the state space X is countable instead of topological or general, the transition kernel is critical in

dealing with recurrence time chains, random walk models and embedded queueing models. In our whole dissertation, we only consider the continuous state space. Irreducibility is a measure of how well a Markovian process communicates. This concept is well used in other properties and important in identifying a Markov chain's behaviors in some situations. We assume throughout that our examples have a transition kernel and are on continuous state spaces.

2.2 Univariate estimation

Consider a probability function π with support $\mathsf{X} \subseteq \mathbb{R}$. In most cases, π is a probability mass function or a probability density function. Suppose $g : \mathsf{X} \to \mathbb{R}$, we want to estimate

$$\theta = E_\pi g = \int_X g(x)\pi(dx).$$

For the $E_\pi g$ estimation, a Markov chain $X = \{X_1, X_2, \dots\}$ is constructed on X with the invariant distribution π through Markov chain Monte Carlo (MCMC) (See e.g. Roberts and Rosenthal (2009)). After simulating X for n steps, we consider the sample average

$$\overline{g}_n := \frac{1}{n}\sum_{i=1}^{n} g(x_i).$$

The strong law of large number (SLLN) says, if $E_\pi|g| < \infty$, then $\overline{g}_n \to E_\pi g$ almost surely when $n \to \infty$. So it is natural to consider using $\overline{g}_n$ to estimate $E_\pi g$.

The Monte Carlo error, $\bar{g}_n - E_\pi g$, can be approximated when a Markov chain CLT holds as follows

$$\sqrt{n}(\bar{g}_n - E_\pi g) \xrightarrow{\mathrm{d}} N(0, \sigma_g^2), \tag{2.1}$$

as $n \to \infty$ where $\sigma_g^2 \in (0, \infty)$. For Markov chains, $\sigma_g^2 \neq Var_\pi g$ due to correlation. If there is an estimator $\hat{\sigma}_n^2 \to \sigma_g^2$, the Monte Carlo Standard Error (MCSE) is $\frac{\hat{\sigma}_n}{\sqrt{n}}$ (Jones, 2004).

Estimators for σ_g need to account for the inherent correlation in the Markov chain, since generally $\sigma_g^2 \neq Var_\pi g$. The most popular technique is the non-overlapping batch means (BM) method (Flegal and Jones, 2010a). For a Markov chain $X = \{X_1, X_2, \ldots, X_n\}$, define $Y_i = g(X_i) - E_\pi g$ for $i = 1, \ldots, n$. Also define $n = a_n b_n$, where a_n is the number of batches and b_n is the batch size. Define $\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$. For $k = 0, \ldots, a_n - 1$, define

$$\bar{Y}_k = \frac{1}{b_n}\sum_{i=1}^{b_n} Y_{kb_n+i}.$$

Then the BM estimator of σ_g^2 is

$$\hat{\sigma}_{BM}^2 = \frac{b_n}{a_n - 1}\sum_{k=0}^{a_n-1}(\bar{Y}_k - \bar{Y}_n)^2.$$

In addition, an asymptotic confidence interval for $E_\pi g$ with half-width

$$t_* \frac{\hat{\sigma}_n}{\sqrt{n}},$$

could be constructed where t_* is a proper quantile. The CLT at (2.1) holds under a variety of conditions, see e.g. Jones (2004).

2.3 Rao-Blackwellization

An alternative estimator could be obtained by averaging conditional expectations. Consider a function of two variables $\pi(x, y)$ and our goal is to estimate the expectation of a function of one variable, $g(x)$. From the Markov chain, we have the following observations: $(X, Y) = \{(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \ldots, \}$. Let $m_Y(y)$ be the marginal density and $f_{X|Y}(x|y)$ be the conditional density. We have

$$E_\pi g = \int\int g(x)\pi(x,y)dxdy = \int \Big[\int g(x) f_{X|Y}(x|y)dx\Big] m_Y(y)dy.$$

Define

$$h(y) = \int g(x) f_{X|Y}(x|y)dx.$$

Then

$$E_\pi g = \int h(y) m_Y(y)dy.$$

By SLLN, note that when $n \to \infty$,

$$\overline{h}_n = \frac{1}{n}\sum_{i=1}^{n} h(y_i) = \frac{1}{n}\sum_{i=1}^{n} \int g(x) f_{X|Y}(x|y_i)dx \xrightarrow{\text{a.s.}} E_\pi g,$$

where $\overline{h}_n$ is called Rao-Blackwellized (RB) estimator of $E_\pi g$ (Casella and Robert, 1996). Our interest in RB estimators is that they motivate a similar thing for density estimation as we will see later.

2.4 Multivariate estimation

In the multivariate situation, π_d is a probability distribution with support $\mathsf{X} \subseteq \mathbb{R}^d$, $d \geq 1$. There is a π_d-integrable function $g : \mathsf{X} \to \mathbb{R}^p$. We are interested in estimating $\theta = E_{\pi_d} g$. For a π_d-invariant Markov chain $\{X_t\}$, an estimate for θ is $\theta_n = \frac{1}{n}\sum_{t=1}^{n} Y_t$, where $\{Y_t\} = \{g(X_t)\}$. $\theta_n \to \theta$ with probability 1 as $n \to \infty$. From the CLT, an approximate sampling distribution for the Monte Carlo error $\theta_n - \theta$ could be derived (Vats et al., 2019), if there is a $p \times p$ positive definite matrix Σ so that when $n \to \infty$,

$$\sqrt{n}\,(\theta_n - \theta) \xrightarrow{\mathrm{d}} N_p(0, \Sigma).$$

It is not difficult to figure out that a large n results in a small Monte Carlo error. The appropriate n could be derived from the univariate CLT

$$\sqrt{n}\,(\theta_{n,i} - \theta_i) \xrightarrow{\mathrm{d}} N(0, \sigma_i^2) \text{ as } n \to \infty,$$

where $\theta_{n,i}$, θ_i and σ_i^2 denote the corresponding ith components of θ_n, θ and ith diagonal element of Σ.

To choose a proper n, the fixed-width sequential stopping rule is suggested in Jones et al. (2006). For a given tolerance ϵ_i from ith component, the simulation process is terminated the first time after $n^* \geq 0$ iterations and satisfy

$$t_* \frac{\sigma_{n,i}}{\sqrt{n}} + \frac{1}{n} \leq \epsilon_i,$$

for all components i, where $\sigma_{n,i}^2$ is a strongly consistent estimator of σ_i^2 and t_* is a proper t-distribution quantile. The proper choice of n^* saves the time effort for running the simulation. However, there are some drawbacks in the application of the fixed-width sequential stopping rule. One drawback is that the essential analysis on the choices of ϵ_i for each $\theta_{n,i}$ requires a lot of work when the number of components p is large. Other difficulties include the delayed termination from conservative rules, the slow mixing components and so on. The relative standard deviation fixed-volume sequential stopping rule is suggested in Vats et al. (2019).

The MCMC basics and point estimation helped us to develop the estimation methods in the following chapters. We will talk more about the MCMC applications later.

Chapter 3

Density Estimation

If we have a random variable $W \sim \pi^d$ and a measurable function $g : X \to \mathbb{R}$, then set $V = g(W)$. We want to estimate the density of V, called f_V or f. For convenience, we assume f is absolutely continuous. In general, it is impossible to calculate f directly. In Bayesian inference, a common problem is how to estimate the marginal posterior densities. The simulation error is routinely omitted from MCMC simulation methods currently. Our study is to assess the simulation error to enhance the reliability of the functional inferences.

One graphical and simple way is using the histogram for density estimation. Under the condition of Markov Chain SLLN, the histogram approximates the true density and is easy to obtain via software.

3.1 Rao-Blackwellized estimator

A better density estimation is the nonparametric density estimate or smoothed histogram. The estimates could derive the corresponding pointwise interval estimates which

may lead to more computational time. Wei and Tanner (1990) provide a parametric density estimator inspired by Rao–Blackwell estimators. Suppose X contain two variables U and V with a target jointed distribution $\pi(u, v)$. The marginal density distributions are m_U and m_V. The Markov chain samples are $\{(U_1, V_1), (U_2, V_2), (U_3, V_3), \ldots, (U_n, V_n)\}$.

The expression for the margin density $m_U(u)$

$$m_U(u) = \int \pi(u, v)dv = \int f_{U|V}(u|v)m_V(v)dv = E_{m_V}[f_{U|V}(u|v)],$$

indicates that m_u could be estimated from Markov chain SLLN. As $n \to \infty$ for each point u,

$$\frac{1}{n}\sum_{i=1}^{n} f_{U|V}(u|V_i) \to m_U(u),$$

where the conditional density $f_{U|V}(u|V_i)$ has to be tractable.

3.2 Kernel density estimation

Kernel density estimation (KDE) is a critical nonparametric density estimation technique (Izenman, 1991). Consider the vector $\hat{f}_n(\overrightarrow{w}) = \left(\hat{f}_n(w_1), \ldots, \hat{f}_n(w_m)\right)$ containing kernel density estimators (Parzen, 1962). Define the kernel density estimator

$$\hat{f}_n(w) = \frac{1}{nh}\sum_{t=1}^{n} K\left(\frac{w - X_t}{h}\right),$$

where K is the kernel. For a fixed h, the expected value of $\hat{f}(w)$ is

$$
\begin{aligned}
E(\hat{f}(w)) &= \frac{1}{nh}\sum_{t=1}^{n} E\left[K\left(\frac{w-X_t}{h}\right)\right] \\
&= \frac{1}{h}E\left[K\left(\frac{w-X_1}{h}\right)\right] \\
&= \frac{1}{h}\int K\left(\frac{w-u}{h}\right) f(u)du \\
&= \int K(z)f(w-zh)dz.
\end{aligned}
$$

It is obvious that $\hat{f}$ is an asymptotic unbiased estimator from the fact that

$$
E(\hat{f}(w)) \to f(w)\int K(z)dz = f(w) \text{ when } h \to 0.
$$

Since the bias adds inaccuracy to estimators and therefore lowers the estimation performance, we need to correct the bias.

If the second derivative f'' of the true density f is absolutely continuous and squared integrable, then a Taylor series to expand $f(w-zh)$ about w is

$$
f(w-zh) = f(w) - hzf'(w) + \frac{1}{2}h^2z^2f''(w) + o(h^2).
$$

It is not difficult to see that the bias of the density estimator is

$$
Bias(\hat{f}(w)) = \frac{h^2}{2}f''(w)u_2(K) + o(h^2), \tag{3.1}
$$

where $u_2(K) = \int x^2K(x)dx > 0$. In the meantime, the variance of the $\hat{f}(w)$ could be calculated as

$$
\begin{aligned}
Var(\hat{f}(w)) &= \frac{1}{nh}\int K^2(z)f(w-hz)dz - \frac{1}{n}\left(E(\hat{f}(w))\right)^2 \\
&= \frac{1}{nh}\int K^2(z)\{f(w)+o(1)\}dz \\
&= \frac{1}{nh}\left(\int K^2(z)dz\right)f(w) + o\left(\frac{1}{nh}\right) \\
&= \frac{1}{nh}R(K)f(w) + o\left(\frac{1}{nh}\right),
\end{aligned}
$$

where $R(g) = \int g^2(z)dz$ for any integrable function g. The Mean Square Error (MSE) is then given by

$$
\begin{aligned}
MSE(\hat{f}(w)) &= Var(\hat{f}(w)) + Bias^2(\hat{f}(w)) \\
&= \frac{1}{nh}R(K)f(w) + \frac{h^4}{4}[f''(w)]^2\mu_2^2(K) + o\left(\frac{1}{nh}\right) + o(h^4).
\end{aligned}
$$

The kernel density estimator given above has the bias which could be corrected, which we discuss later.

The problem of estimating the r'th derivative of the density $f(w)$ is

$$
f^{(r)}(w) = \frac{d^r}{dw^r}f(w).
$$

It is natural to use the estimator by taking derivatives of the kernel density estimator. The form is

$$
\hat{f}^{(r)}(w) = \frac{d^r}{dw^r}\hat{f}(w) = \frac{1}{nh^{1+r}}\sum_{t=1}^{n} K^{(r)}\left(\frac{w-X_t}{h}\right),
$$

where

$$K^{(r)}(w) = \frac{d^r}{dw^r}K(w).$$

This estimator is valid for use if $K^{(r)}(w)$ exists and is non-zero. A common choice is Guassian kernel since it has derivatives of all orders.

3.3 Kernels

Epanechnikov (1969) talked about the application of kernels in estimating densities. The kernel refers to a smooth function K such that

$$K(x) \geq 0, \int K(x)dx = 1, \int xK(x)dx = 0, \text{and } \int x^2K(x)dx > 0.$$

The main purpose for using kernels is to get the local average from weighting. The weighting is based on the distance from each x_i to the point x where we want to estimate the function.

Some kernels are common for use such as

$$\text{boxcar kernel: } K(x) = \frac{1}{2}I(x),$$

$$\text{Gaussian kernel: } K(x) = \frac{1}{\sqrt{2\pi}}exp\left(-\frac{x^2}{2}\right),$$

$$\text{Epanechnikov kernel: } K(x) = \frac{3}{4}(1-x^2)I(x),$$

$$\text{tricube kernel: } K(t) = \frac{70}{81}(1-|t|^3)^3I(t),$$

where

$$I(t) = \begin{cases} 1 & \text{if } |t| \leq 1; \\ 0 & \text{if } |t| > 1. \end{cases}$$

We will show later that the estimation depends on the choice of h, the bandwidth that indicates that which points should be included in the computation for the local average.

3.4 Bias-reducing kernel

Jones et al. (1995) discussed their way in correcting bias in kernel density estimation. While their method is a simple, two-stage multiplicative bias correction, the method is not efficient in calculating the batch estimators and the covariance matrix.

The bias correction procedure is also discussed in the kernel density estimation with Markov chain settings. Calonico et al. (2018) discussed the bias corrected kernel in the density estimation. While there are widely-stated studies about the bias correction in the kernel density estimation, we fill in the gap for applying the bias correction procedure together with MCMC and covariance matrix estimation.

Instead of using bias-correction technique on the kernel, another method for bias correction is applying the bias-reducing kernels. The order of a kernel, v is defined as the order of the first non-zero moment. For example, a Guassian kernel

$$k_{2,\phi}(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right), \tag{3.2}$$

is second-order kernel.

A kernel is high-order kernel if $v > 2$. These kernels will have negative parts and are not probability densities and are referred as bias-reducing kernels. For example, fourth-order Guassian kernel is

$$k_{4,\phi}(u) = \frac{1}{2}(3 - u^2)k_{2,\phi}(u).$$

For a kernel density estimator

$$\hat{f}_{n,old}(w) = \frac{1}{nh}\sum_{t=1}^{n} k_{2,\phi}\left(\frac{w - X_t}{h}\right), \tag{3.3}$$

with a 2nd order normal kernel $k_{2,\phi}(u)$, the 2nd order derivative of (3.3) is

$$\hat{f}^{(2)}_{n,old}(w) = \frac{1}{nh^3}\sum_{t=1}^{n} k^{(2)}_{2,\phi}\left(\frac{w - X_t}{h}\right). \tag{3.4}$$

And we could derive

$$k^{(2)}_{2,\phi}(u) = (u^2 - 1)k_{2,\phi}(u),$$

so the estimator with corrected bias from (3.4) is

$$
\begin{aligned}
\hat{f}_n(w) &= \hat{f}_{n,old}(w) - Bias(\hat{f}(w)) \\
&= \hat{f}_{n,old}(w) - \left(\frac{h^2}{2} \hat{f}^{(2)}_{n,old}(w) u_2(k_{2,\phi}) + o(h^2) \right) \\
&= \frac{1}{nh} \sum_{t=1}^{n} k_{2,\phi} \left(\frac{w - X_t}{h} \right) - \left(\frac{h^2}{2} \frac{1}{nh^3} \sum_{t=1}^{n} k^{(2)}_{2,\phi} \left(\frac{w - X_t}{h} \right) \right) \\
&= \frac{1}{nh} \sum_{t=1}^{n} k_{2,\phi} \left(\frac{w - X_t}{h} \right) - \left(\frac{1}{2nh} \sum_{t=1}^{n} (w^2 - 1) k_{2,\phi} \left(\frac{w - X_t}{h} \right) \right) \\
&= \frac{1}{nh} \sum_{t=1}^{n} (1 - \frac{1}{2}(w^2 - 1)) k_{2,\phi} \left(\frac{w - X_t}{h} \right) \\
&= \frac{1}{nh} \sum_{t=1}^{n} \frac{1}{2}(3 - w^2)) k_{2,\phi} \left(\frac{w - X_t}{h} \right).
\end{aligned}
$$

The similar bias-corrected kernel was discussed (Jones et al., 1995).

The bias-corrected estimator is

$$
\hat{f}_n(w) = \frac{1}{nh} \sum_{t=1}^{n} k_{4,\phi} \left(\frac{w - X_t}{h} \right). \tag{3.5}
$$

For convenience, the density estimator we used is always referred to this debiased estimator $\hat{f}_n(w)$ if not particularly indicated.

3.4.1 AR(1) model example

Consider the AR(1) process defined by

$$X_t = \phi X_{t-1} + \epsilon_t \text{ for t=1,...,n,} \tag{3.6}$$

where $\epsilon_t \sim N(0,1)$. The true distribution for X_t is $N(0, \frac{1}{1-\phi^2})$. For 200 replications and the confidence level $1-\alpha = 90\%$. The MCMC sample length is $n = 400,000$. The number of batches is $a_n = 400$ and the batch size is $b_n = 1000$. The densities are estimated from -10 to 10 on $m = 51$ points equally spaced with $\phi = 0.95$. The bandwidth is default $h = 0.3$. We used the pointwise coverage to compare different kernels.

Table 3.1: Pointwise coverage comparisons for AR(1) process using different kernels.

Kernel	4th order normal kernel	2nd order normal kernel
Average pointwise coverage	0.887	0.863

From Table 3.1, the debiased (4th order) kernel works better than the normal kernel regarding the average pointwise coverage. We will talk more about this example using the bandwidth selection later in this chapter.

3.5 Bias and variance for the estimator

The bias and variance for our estimator could be derived (Cheng and Chen, 2019).

Lemma 1(Pointwise bias): The bias of $\hat{f}_n(w)$ in (3.5) is

$$E(\hat{f}_n(w) - f(w)) = O(h^{2+\delta_0}).$$

The variance is

$$Var(\hat{f}_n(w)) = O\left(\frac{1}{nh}\right).$$

Proof. Recall

$$\begin{aligned}\hat{f}_n(w) &= \frac{1}{nh}\sum_{t=1}^{n} k_{4,\phi}\left(\frac{w - X_t}{h}\right) \\ &= \frac{1}{nh}\sum_{t=1}^{n} M\left(\frac{w - X_t}{h}\right).\end{aligned}$$

Then we have

$$\begin{aligned}E(\hat{f}_n(w)) &= f(w) + \frac{h^2}{2}u_2(k_{2,\phi})E(f^{(2)}(w)) + O(h^{2+\delta_0}) - \frac{h^2}{2}u_2(k_{2,\phi})(f^{(2)}(w) + O(h^{\delta_0})) \\ &= f(w) + O(h^{2+\delta_0} + h^2 \cdot h^{\delta_0}) \\ &= f(w) + O(h^{2+\delta_0}).\end{aligned}$$

For the variance part

$$\begin{aligned}Var(\hat{f}_n(w)) &= \sum_{t=1}^{n} Var\left[M\left(\frac{w - X_t}{h}\right)\right] + 2\sum_{i<j} Cov\left[M\left(\frac{w - X_i}{h}\right), M\left(\frac{w - X_j}{h}\right)\right] \\ &= L_1 + L_2,\end{aligned}$$

where $\sum_{i<j} = \sum_{j=1}^{n}\sum_{i=1}^{j-1}$ is the double sum.

By parts, we have

$$\begin{aligned}
L_1 &= nVar\left[M\left(\frac{w-X_t}{h}\right)\right] \\
&= \frac{1}{nh^2}\left\{E\left[M\left(\frac{w-X_t}{h}\right)^2\right] - E\left[M\left(\frac{w-X_t}{h}\right)\right]^2\right\} \\
&= \frac{1}{nh^2}\left\{h\int f(w-zh)M^2(z)dz - \left(h\int f(w-zh)M(z)dz\right)^2\right\} \\
&= \frac{1}{nh}\left(f(w)\int M^2(z)dz + O(h^2) + O(h)\right) \\
&= O\left(\frac{1}{nh}\right),
\end{aligned}$$

$$\begin{aligned}
L_2 &= 2\sum_{i<j} r_{i,j} Var\left[M\left(\frac{w-X_t}{h}\right)\right] \\
&= \left\{Var\left[M\left(\frac{w-X_t}{h}\right)\right]\right\}\left(2\sum_{i<j} r_{i,j}\right) \\
&= \left\{nVar\left[M\left(\frac{w-X_t}{h}\right)\right]\right\}\left(\frac{2}{n}\sum_{i<j} r_{i,j}\right) \\
&= (L_1)\left(\frac{2}{n}\sum_{i<j} r_{i,j}\right) \\
&= O\left(\frac{1}{nh}\right),
\end{aligned}$$

where $r_{i,j}$ is the correlation $Cov\left(M\left(\frac{w-X_i}{h}\right), M\left(\frac{w-X_j}{h}\right)\right)$.

The last step in the previous equation requires an assumption

$$\frac{2}{n}\sum_{i<j} r_{i,j} = O(1),$$

Or equivalently

$$\sum_{i<j} r_{i,j} = O(n).$$

This assumption is guaranteed from the finite covariance matrix Σ. ■

3.6 Bandwidth choice

When it comes to the bandwidth selection in the kernel density estimation, Kim et al. (2016) discussed KDE with bandwidth selection with MCMC setting. Sheather and Jones (1991) stated various ways in selecting the bandwidth in the kernel density estimation.

A kernel-based estimate of mean integrated squared error (MISE) could be obtained by the asymptotic expansion (AMISE) from Sheather and Jones (1991):

$$AMISE(h) = \frac{1}{nh} R(K) + \frac{1}{4} h^4 u_2^2(K) R(f''),$$

where $R(g) = \int g^2(x)dx$ and $u_2(K) = \int z^2 K(z)dz > 0$.

The objective function is

$$\psi(h) = \frac{1}{nh} R(K) + \frac{1}{4} h^4 u_2^2(K) \hat{S}(\beta),$$

where $\hat{S}(\beta)$ is the kernel-based estimate of $R(f'')$ using an appropriate bandwidth β. The estimation of f'' is given by

$$\hat{f}_n^{(2)}(w) = \frac{1}{n\beta^3} \sum_{t=1}^{n} k_{4,\phi}^{(2)} \left(\frac{w - X_t}{\beta} \right).$$

If β does not depend on h, the minimization of ψ is given by

$$\tilde{h} = \left[\frac{R(K)}{u_2^2(K)\hat{S}(\beta)} \right]^{\frac{1}{5}} n^{-\frac{1}{5}}.$$

To illustrate the bandwidth selection effect, we use the AR(1) example (3.6). We compare the pointwise coverage with default $h = 0.3$ and optimal $h = 0.162$.

Table 3.2: Pointwise coverage comparisons for AR(1) model using default and selected bandwidth h.

h	0.3	0.162
Average pointwise coverage	0.887	0.907

The average pointwise coverage from the optimal bandwidth h is larger than the coverage with default $h = 0.3$. We concluded that the bandwidth selection did improve the coverage probabilities.

In the later chapters, we will show that the pointwise bands fail to cover the functions simultaneously and therefore introduce our simultaneous confidence bands.

Chapter 4

General Function Estimation

For a general function $f(w)$, the estimation procedure is quite different while the confidence band could be obtained similar to the kernel density estimator. We consider a functional estimator $\hat{f}$ with a certain domain in $\mathbb{R}$. Again, suppose f is estimated at m points on a compact set $D = [a, b]$. $\overrightarrow{w} = (w_1, \ldots, w_m)^T$ with $a = w_1 < w_2 < \cdots < w_m = b$. The vector estimator $\hat{f}_n(\overrightarrow{w}) = \left(\hat{f}_n(w_1), \ldots, \hat{f}_n(w_m)\right)$ could be obtained via parametric and nonparametric approaches (Wei and Tanner, 1990; Wasserman, 2006).

Functional estimation is more general than density estimation. First, the functions are arbitrary and have fewer assumptions. For example, densities need to be non-negative and integrated to 1. In this way, functions could have more unexpected properties (saddle points, for example). Further, there are more model-based methods in functional estimation, such as local linear regression, longitudinal model, and others. These methods provide a baseline for functional estimation with different pros and cons. Besides these differences, there is an increasing demand in evaluating functional estimators with competitive methods.

A general method for functions is more critical in analysis and research than a density-limited method.

The similarities density estimation and functional estimation between are still alive The findings in the density estimation would be transported to functions without too much effects with little obstacles for the notations and definitions. Theorems and techniques like CLT, Metropolis-Hastings algorithm and random walk chains are also useful. With all these challenges and opportunities, we proposed a unique methodology to build simultaneous confidence bands for functional estimators.

4.1 Local linear regression

Local linear regression is a well-defined and popular method in the functional estimation (Hastie et al., 2009). For a set of $(x_i, y_i), i = 1, ..., n$, we consider the following way to construct the estimate for the target function $y = f(x)$. For an arbitrary x_0, the point we want to estimate, consider a linear estimate

$$\hat{f}(x_0) = \alpha(x_0) + \beta(x_0)x_0,$$

where $\alpha(x_0)$ and $\beta(x_0)$ are parameters based on x_0. The calculation is based on the minimization

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{n} K_\lambda(x_0, x_i)(y_i - \alpha(x_0) - \beta(x_0)x_i),$$

where

$$K_\lambda(x_0, x) = D\left(\frac{|x_0 - x|}{\lambda}\right),$$

and

$$D(t) = \begin{cases} \frac{3}{4}(1-t^2) & \text{if } |t| < 1; \\ 0 & \text{otherwise.} \end{cases}$$

The choice of λ is between 0 and 1 with default 0.15. The function estimate $f(x_0)$ is given from $\alpha(x_0)$ and $\beta(x_0)$ by the minimization process.

The matrix form of the minimization is plausible. Define the vector $b(x)^T = (1, x)$. Suppose B is the $n \times 2$ matrix with ith row $b(x_i)^T$ and $W(x_0)$ is the $n \times n$ diagonal matrix with ith diagonal element $K_\lambda(x_0, x_i)$. So the estimator

$$\begin{aligned} \hat{f}(x_0) &= b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0) \overrightarrow{y} \\ &= \sum_{i=1}^{n} l_i(x_0) y_i, \end{aligned} \tag{4.1}$$

where the weights $l_i(x_0)$ are the mixture of the weighting kernel $K_\lambda(x_0, \cdot)$ and the least squares operations.

4.2 Bias

In (4.1), the constraints are $\sum_{i=1}^{n} l_i(x_0) = 1$ and $\sum_{i=1}^{n}(x_i - x_0)l_i(x_0) = 0$. With Taylor expansion, the expected value

$$\begin{aligned} E\hat{f}(x_0) &= \sum_{i=1}^{n} l_i(x_0) f(x_i) \\ &= f(x_0)\sum_{i=1}^{n} l_i(x_0) + f'(x_0)\sum_{i=1}^{n}(x_i - x_0)l_i(x_0) + f''(x_0)\sum_{i=1}^{n}(x_i - x_0)^2 l_i(x_0) + R \\ &= f(x_0) + f''(x_0)\sum_{i=1}^{n}(x_i - x_0)^2 l_i(x_0) + R, \end{aligned} \tag{4.2}$$

where the remainder term R involves third and higher order derivatives and could be omitted for convenience. On the computing aspect, with *hessian* function in R, the $f''(x_0)$ could be estimated for each point x_0. Hence we could estimate the $f(x)$ with bias correction.

The bias-corrected estimator we use throughout this dissertation is

$$\hat{f}(x_0) = \sum_{i=1}^{n} l_i(x_0) y_i - \hat{f}''(x_0)\sum_{i=1}^{n}(x_i - x_0)^2 l_i(x_0), \tag{4.3}$$

where $\hat{f}''(x_0)$ is an estimator for $f''(x_0)$ The details to obtain this estimator are in the appendix at the end of this chapter.

4.3 Example

Consider the function

$$f(x) = x + xsin(x),$$

where $x's$ are simulated from $Uniform(-2.5, 2.5)$ (using random walk to add correlations to the draws). The function values are generated through $y_i = f(x_i) + \epsilon_i$ and the noise is $\epsilon_i \sim N(0, 0.5^2)$. The number of points to be estimated is $m \in \{40, 80, 120\}$ which are equally spaced between $(-2, 2)$. The bandwidth for the kernel $K_\lambda(x_0, x)$ is $\lambda = 0.15$. In each simulation, $n = 20,000$ draws were generated with the number of batches $a_n = 100$ and the batch size $b_n = 200$. The confidence level is 90%. The replicates are 200.

We compared the pointwise coverage probabilities through bias-corrected estimator (4.3) and estimator without bias-correction (4.1).

Table 4.1: Pointwise coverage for the function $x + xsin(x)$ example with and without bias correction.

m	40	80	120
With bias correction	0.943	0.951	0.953
Without bias correction	0.182	0.183	0.178

The example showed that our bias-corrected estimator achieved higher pointwise coverage probabilities than the one without bias-correction with different m choices. So we would use the bias-corrected estimator (4.3) afterwards.

Later we will see though the pointwise coverage looks fine even for the bias-corrected estimator, the simultaneous coverage is beyond the desired coverage. We will talk about more details and other confidence bands methods for this example.

4.4 Appendix: functional estimator with bias correction

Recall the estimator (4.1) and the bias in (4.2). There are two steps to directly estimate the second derivative $f''(x_0)$. The first step is to derive the first and second derivatives of $K_\lambda(x_0, x)$ regarding x_0 for arbitrary x. The second step is to find the closed form of $f''(x_0)$.

First we will get the derivatives of $K_\lambda(x_0, x_i)$. Recall

$$K_\lambda(x_0, x) = D\left(\frac{|x_0 - x|}{\lambda}\right),$$

and

$$D(t) = \begin{cases} (1 - |t|^3)^3 & \text{if } |t| < 1; \\ 0 & \text{otherwise.} \end{cases}$$

So the combined form for the kernel $K_\lambda(x_0, x)$

$$K_\lambda(x_0, x) = \begin{cases} \left(1 - \frac{|x - x_0|^3}{\lambda^3}\right)^3 & \text{if } |x - x_0| < 1; \\ 0 & \text{otherwise.} \end{cases}$$

The first derivative is

$$K'_\lambda(x_0, x) = \frac{dK_\lambda(x_0, x)}{dx_0} = \begin{cases} 9[1 + \frac{(x_0 - x)^3}{\lambda^3}]^2 \frac{(x_0 - x)^2}{\lambda^3} & \text{if } x - \lambda < x_0 \leq x; \\ -9[1 - \frac{(x_0 - x)^3}{\lambda^3}]^2 \frac{(x_0 - x)^2}{\lambda^3} & \text{if } x < x_0 < x + \lambda; \\ 0 & \text{otherwise.} \end{cases}$$

The second derivative is

$$K''_\lambda(x_0,x)=\frac{dK'_\lambda(x_0,x)}{dx_0}=\begin{cases}18(\frac{x_0-x}{\lambda^3})[1+\frac{(x_0-x)^3}{\lambda^3}][1+\frac{4(x_0-x)^3}{\lambda^3}] & \text{if } x-\lambda<x_0\le x;\\ 18(\frac{x_0-x}{\lambda^3})[1-\frac{(x_0-x)^3}{\lambda^3}][-1+\frac{4(x_0-x)^3}{\lambda^3}] & \text{if } x<x_0<x+\lambda;\\ 0 & \text{otherwise.}\end{cases}$$

Then we rewrite $f(x_0)$ in terms of x_0. Recall our estimator

$$\hat{f}(x_0)=b(x_0)^T(B^TW(x_0)B)^{-1}B^TW(x_0)\overrightarrow{y}.$$

In the estimation formula,

$$b(x_0)^T=\begin{pmatrix}1 & x_0\end{pmatrix},$$

$$B=\begin{pmatrix}1 & x_1\\ 1 & x_2\\ \vdots & \vdots\\ 1 & x_n\end{pmatrix}=(\overrightarrow{1},\overrightarrow{x}),$$

where the vector $\overrightarrow{x}=(x_1,x_2,\ldots,x_n)^T$. For our convenience, the diagonal matrix of the kernel $K_\lambda(x_0,x_i)$'s is

$$W(x_0)=\begin{pmatrix}K_1 & & & \\ & K_2 & & \\ & & \ddots & \\ & & & K_n\end{pmatrix},$$

where K_i is a simplified version for $K_\lambda(x_0, x_i)$ which yields

$$B^T W(x_0) = \begin{pmatrix} K_1 & K_2 & \cdots & K_n \\ x_1 K_1 & x_2 K_2 & \cdots & x_n K_n \end{pmatrix}.$$

Then

$$B^T W(x_0) B = \begin{pmatrix} \sum K & \sum xK \\ \sum xK & \sum x^2 K \end{pmatrix},$$

where $\sum K = \sum_{i=1}^n K_i$, $\sum xK = \sum_{i=1}^n x_i K_i$ and $\sum x^2 K = \sum_{i=1}^n x_i^2 K_i$.

So the matrix multiplication inverse is

$$(B^T W(x_0) B)^{-1} = \frac{1}{(\sum K)(\sum x^2 K) - (\sum xK)^2} \begin{pmatrix} \sum x^2 K & -\sum xK \\ -\sum xK & \sum K \end{pmatrix}.$$

The vector $\overrightarrow{y} = (y_1, y_2, \ldots, y_n)^T$. We now have

$$B^T W(x_0) \overrightarrow{y} = \begin{pmatrix} \sum yK \\ \sum xyK \end{pmatrix},$$

where $\sum yK = \sum_{i=1}^n y_i K_i$ and $\sum xyK = \sum_{i=1}^n x_i y_i K_i$.

We combine the formula for $b(x_0)^T$, $(B^T W(x_0) B)^{-1}$ and $B^T W(x_0) \overrightarrow{y}$ to get the estimator

$$\begin{aligned} \hat{f}(x_0) &= b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0) y \\ &= \frac{1}{Z}(J + L), \end{aligned}$$

where

$$Z = (\sum K)(\sum x^2K) - (\sum xK)^2,$$

$$J = (\sum x^2K)(\sum yK) - (\sum xyK)(\sum xK),$$

$$L = x_0[(\sum K)(\sum xyK) - (\sum yK)(\sum xK)],$$

We can derive the estimator for the first derivative $f'(x_0)$

$$\hat{f}'(x_0) = -\frac{Z'}{Z^2}(J + L) + \frac{1}{Z}(J' + L'),$$

where

$$Z' = (\sum K')(\sum x^2K) + (\sum K)(\sum x^2K') - 2(\sum xK)(\sum xK'),$$

$$J' = (\sum x^2K')(\sum yK) + (\sum x^2K)(\sum yK') - (\sum xyK')(\sum xK) - (\sum xyK)(\sum xK'),$$

$$L' = (\sum K)(\sum xyK) - (\sum yK)(\sum xK) + x_0[(\sum K')(\sum xyK)$$
$$+ (\sum K)(\sum xyK') - (\sum yK')(\sum xK) - (\sum yK)(\sum xK')].$$

Furthermore, we get the second derivative $f''(x_0)$ estimator

$$\hat{f}''(x_0) = \frac{2(Z')^2 - Z^2Z''}{Z^4}(J + L) - \frac{2Z'}{Z^2}(J' + L') + \frac{1}{Z}(J'' + L''),$$

where

$$Z'' = (\sum K'')(\sum x^2K) + 2(\sum K')(\sum x^2K') + (\sum K)(\sum x^2K'')$$
$$- 2(\sum xK')^2 + (\sum xK)(\sum xK''),$$

$$J'' = (\sum x^2K'')(\sum yK) + 2(\sum x^2K')(\sum yK') + (\sum x^2K)(\sum yK'') - (\sum xyK'')(\sum xK)$$
$$- 2(\sum xyK')(\sum xK') - (\sum xyK)(\sum xK''),$$

$$L'' = 2[(\sum K')(\sum xyK) + (\sum K)(\sum xyK') - (\sum yK')(\sum xK) - (\sum yK)(\sum xK')]$$
$$+ x_0 2[(\sum K'')(\sum xyK) + 2(\sum K')(\sum xyK') + (\sum K)(\sum xyK'') - (\sum yK'')(\sum xK)$$
$$- 2(\sum yK')(\sum xK') - (\sum yK)(\sum xK'')].$$

After the second derivative, we have the bias-corrected estimator:

$$\hat{f}(x_0) - \hat{f}''(x_0)\sum_{i=1}^{n}(x_i - x_0)^2 l_i(x_0)$$
$$= \sum_{i=1}^{n} l_i(x_0)y_i - \hat{f}''(x_0)\sum_{i=1}^{n}(x_i - x_0)^2 l_i(x_0).$$

Chapter 5

Central Limit Theorem and Covariance Matrix Estimation

This chapter introduces the covariance matrix Σ estimation with the prerequisites. The matrix Σ measures the variances for each point with its diagonal values and the covariances between two points with its off-diagonal values.

We consider the estimation part first. Recall the estimator $\hat{f}$ for f with the Monte Carlo sample $X_1, \ldots, X_n$. Again, suppose f is estimated at m points on a compact set $D = [a, b]$. The estimated vector is $\overrightarrow{w} = (w_1, \ldots, w_m)^T$ with $a = w_1 < y_2 < \cdots < w_m = b$. The estimation methods are separated for the density functions and the general functions.

5.1 Estimators for functions

We will summarize the estimators for densities and general functions. First consider the estimation of a density f. The estimator we use is

$$\hat{f}_n(w) = \frac{1}{nh}\sum_{t=1}^{n} k_{4,\phi}\left(\frac{w - X_t}{h}\right),$$

where $k_{4,\phi}$ is the 4th order Guassian kernel and bandwidth h is selected from an optimal procedure.

We now consider general functional estimation. The estimator is

$$\hat{f}_n(w) = \sum_{i=1}^{n} l_i(w)X_i - \hat{f}''(w)\sum_{i=1}^{n}(X_i - w)^2 l_i(w), \tag{5.1}$$

where the weights $l_i(w)$ are the mixture of the weighting kernel and the least squares operations at each X_i. The $\hat{f}''(w)$ is the estimation for the second-order derivative $f''(w)$.

5.2 Central limit theorem

The Central Limit Theorem (CLT) was discussed with Monte Carlo settings before (Jones, 2004). For kernel density estimation, we have the following theorem.

Theorem 1 *Let $X = \{X_1, X_2, \ldots\}$ be a Harris ergodic Markov chain from a probability distribution f having support $\mathbb{X}$. Suppose the corresponding cdf $F(w)$ is absolutely continuous and twice-differentiable. The density function f satisfies $0 < f(\eta) < \infty$ and the first derivative $f'(w)$ is bounded in some neighborhood of η. For the following kernel estimator at any point w:*

$$\hat{f}_n(w) = \frac{1}{nh}\sum_{t=1}^{n} k_{4,\phi}\left(\frac{w - X_t}{h}\right),$$

if the bandwidth h is fixed, then the following CLT holds as $n \to \infty$

$$\sqrt{n}(\hat{f}_n - f) \xrightarrow{d} \mathcal{G}(0, \Sigma),$$

where $\mathcal{G}(0, \Sigma)$ *denotes a Gaussian process with mean zero and covariance* Σ*.*

For the general function estimation, we have the following remark.

Remark 2 *Let* $(X, Y) = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ *be a Harris ergodic Markov chain with a target distribution* $y = f(x)$ *having support* $\mathbb{X}$*. The function* f *satisfies* $f(\eta) < \infty$ *and the first derivative* $f'(w)$ *is bounded in some neighborhood of* η*. For the following function estimator at any point* w*:*

$$\hat{f}_n(w) = \sum_{i=1}^{n} l_i(w) X_i - \hat{f}''(w) \sum_{i=1}^{n} (X_i - w)^2 l_i(w),$$

in (5.1)*, if the bandwidth* λ *from the weighting kernel* K_λ *is fixed, the following CLT holds as* $n \to \infty$

$$\sqrt{n}(\hat{f}_n - f) \xrightarrow{d} \mathcal{G}(0, \Sigma),$$

where $\mathcal{G}(0, \Sigma)$ *denotes a Gaussian process with mean zero and covariance* Σ*.*

5.3 Covariance matrix estimation

We apply the batch mean method to estimate the covariance matrix Σ (Cheng and Chen, 2019; Politis et al., 1999; Vats et al., 2019, 2020).

Denote $n = a_n b_n$, where a_n is the number of batches and b_n is the corresponding batch size. For density estimations, the batch estimator $\tilde{f}_k(w)$ for batch k is

$$\tilde{f}_k(w) = \frac{1}{b_n h} \sum_{t=1}^{b_n} k_{4,\phi} \left(\frac{w - X_{kb_n+t}}{h} \right), \text{for } k = 0, \ldots, a_n - 1.$$

For general function estimation, the batch estimator $\tilde{f}_k(w)$ for batch k is

$$\tilde{f}_k(w) = \sum_{t=1}^{b_n} l_{kb_n+t}(w) X_{kb_n+t} - \hat{f}''(w) \sum_{t=1}^{b_n} (X_{kb_n+t} - w)^2 l_{kb_n+t}(w), \text{for } k = 0, \ldots, a_n - 1.$$

These yield the batch estimator $\tilde{f}_k(\overrightarrow{w}) = \left(\tilde{f}_k(w_1), \ldots, \tilde{f}_k(w_m) \right)$ for the batch k. A batch mean estimator of Σ is

$$\hat{\Sigma} = \frac{b_n}{a_n - 1} \sum_{k=0}^{a_n - 1} \left(\tilde{f}_k(\overrightarrow{w}) - \hat{f}_n(\overrightarrow{w}) \right) \left(\tilde{f}_k(\overrightarrow{w}) - \hat{f}_n(\overrightarrow{w}) \right)^T. \tag{5.2}$$

The estimator $\hat{\Sigma}$ requires the bias correction not only overall but also within each batch, which captures the variability from both the estimation of the function and the estimation of the bias.

For a fixed sample size n, we should choose the number of batches a_n and the batch size b_n to satisfy $n = a_n b_n$. This requirement is from the MCMC sample assumptions. Once we pick the value for a_n, b_n is then fiexed.

If a_n is too large, the value of b_n is small so we do have enough batch data to generate the accurate batch estimators. If b_n is too large, the value for a_n is small, and hence the variability of the estimator will be high. We use $a_n = \lfloor \sqrt{\frac{n}{2.5}} \rfloor$ and $b_n = \left\lfloor \frac{n}{a_n} \right\rfloor$ as the referenced values.

Chapter 6

Confidence Intervals and Bands

To build confidence bands for densities and functions, we consider pointwise, Bonferroni and simultaneous approaches. Pointwise bands are widely used to build an interval for a functional estimate at a single point. While these are appropriate in measuring the uncertainty at a point, pointwise bands lack the ability to capture the function values for all the points where we are interested. Bonferroni bands are the multiplicity corrected version of pointwise bands by considering the number of points to be estimated. However, Bonferroni bands are conservative when there are a large number of points to be estimated. Simultaneous bands lie between pointwise and Bonferroni methods. These methods require simulations to determine the lower and upper bounds to achieve the desired confidence level.

6.1 Pointwise bands

Constructing pointwise confidence bands is straightforward. We are looking to find the lower and upper bounds (L_p, U_p) for the function f and the estimated vector

$\overrightarrow{w} = (w_1, \ldots, w_m)^T$ such that

$$P(L_p(w_i) \leq f(w_i) \leq U_p(w_i)) = 1 - \alpha \text{ for each i=1,...,m.}$$

Suppose $\hat{f}(w)$ as an estimator for a function $f(w)$ at any point w. The univariate CLT at w is

$$\sqrt{n}(\hat{f}_n(w) - f(w)) \to N(0, \sigma_f^2(w)) \text{ as } n \to \infty,$$

where $N(0, \sigma_f^2(w))$ is a normal distribution with mean 0 and the standard deviation $\sigma_f(w)$ (the standard deviation at w). A $100(1-\alpha)\%$ pointwise confidence interval for $f(w)$ at point w is

$$\left(\hat{f}_n(w) - z_{\frac{\alpha}{2}} \frac{\hat{\sigma}_n(w)}{\sqrt{n}}, \hat{f}_n(w) + z_{\frac{\alpha}{2}} \frac{\hat{\sigma}_n(w)}{\sqrt{n}}\right),$$

where $z_{\frac{\alpha}{2}}$ is the upper $\frac{\alpha}{2}$ quantile of the standard normal distribution z and $\hat{\sigma}_n(w)$ is a proper estimate for $\sigma_f(w)$.

For w_i in the estimated vector $\overrightarrow{w} = (w_1, \ldots, w_m)^T$

$$\sqrt{n}(\hat{f}_n(w_i) - f(w_i)) \to N(0, \Sigma_{ii}) \text{ as } n \to \infty,$$

where $\hat{f}_n(w_i)$, $f(w_i)$ and Σ_{ii} denote the corresponding ith components of $\hat{f}_n(w)$, $f(w)$ and ith diagonal element of Σ.

Recall the pointwise standard error for $\hat{f}_n(w_i)$ is

$$\hat{\sigma_i} = \frac{1}{\sqrt{n}} \sqrt{\hat{\Sigma}_{ii}},$$

where $\hat{\Sigma}_{ii}$ is the ith diagonal element of $\hat{\Sigma}$.

A $100(1-\alpha)\%$ pointwise confidence interval for ith component, $f(w_i)$ of $f(\overrightarrow{w})$ is

$$(L_p(w_i), U_p(w_i)) = \left(\hat{f}_n(w_i) - z_{\frac{\alpha}{2}}\hat{\sigma_i}, \hat{f}_n(w_i) + z_{\frac{\alpha}{2}}\hat{\sigma_i}\right).$$

So the pointwise band is the Cartesian product of these pointwise confidence intervals. Note the overall coverage is

$$P(L_p(w_i) \leq f(w_i) \leq U_p(w_i)) \text{ for all i=1,...,m}) \leq 1 - \alpha.$$

The equality holds when all the events $L_p(w_i) \leq f(w_i) \leq U_p(w_i)$'s are identical or $m = 1$. The overall coverage decreases as m increases.

6.2 Bonferroni bands

The Bonferroni band is obtained from a Bonferroni multiple comparisons adjustment to the pointwise band. Dunn (1958) applied the Bonferroni inequalities in building confidence intervals to develop the Bonferroni band. We are looking to find the lower and upper bounds (L_b, U_b) for the function f and the estimated vector $\overrightarrow{w} = (w_1, \ldots, w_m)^T$ such that

$$P(L_b(w_i) \leq f(w_i) \leq U_b(w_i)) = 1 - \frac{\alpha}{m} \text{ for each i=1,...,m.}$$

For m points, a $100(1-\alpha)\%$ Bonferroni confidence interval for ith component, $f(w_i)$ of $f(\overrightarrow{w})$ is

$$(L_b(w_i), U_b(w_i)) = \left(\hat{f}_n(w_i) - z_{\frac{\alpha}{2m}}\hat{\sigma}_i, \hat{f}_n(w_i) + z_{\frac{\alpha}{2m}}\hat{\sigma}_i\right),$$

where $z_{\frac{\alpha}{2m}}$ is the upper $\frac{\alpha}{2m}$ quantile of the standard normal distribution z. The overall coverage is

$$P(L_b(w_i) \leq f(w_i) \leq U_b(w_i) \text{ for all i=1,...,m}) \geq 1 - \alpha.$$

The equality holds when the sample are independent or $m = 1$. The overall coverage increases as m increases.

6.3 Pointwise confidence bands versus simultaneous confidence bands

From any methodology, we find the $(1-\alpha)$ confidence intervals for $f(y_i)$ is (L_i, U_i). From the pointwise method, suppose we could achieve $(1-\alpha)$ confidence for each point w_i. However, the coverage is called "pointwise", which means that the confidence we achieve is only for a certain point.

In some cases, we want to build simultaneous confidence band for a function $f(w)$ on m points with bands $(L_s(w_i), U_s(w_i))$ that there are approximately $(1-\alpha)r$ times that

$$f(w_i) \in (L_s(w_i), U_s(w_i)) \text{ for every i,}$$

if we repeat building the confidence bands for r times. For these bands, we call them simultaneous confidence bands.

Simultaneous confidence bands are gaining more importance recently, see e.g. Montiel Olea and Plagborg-Møller (2019). Pointwise confidence bands are inferior in achieving the whole evaluation in a certain estimator. An estimator that could contribute to a good coverage behaviour may be proper in some situations when we focus on certain points.

For observational functions changing over time, for example, a stock price, the situation is quite different. For stockholders, they want to know not only the price after 3 months but also the price in 1 month, 2 months up to 36 months. If the expected price

increases dramatically in a certain month but they fail to recognize, they will lose the opportunity to sell them and gain profits. Another example is temperature where meteorologists need to monitor temperature in order to help predict harm that a severe temperature condition could provide. For example, a dramatic increase or drop in temperature would harm the human activities and cause economic losses. However, if meteorologists could not get an idea about how the temperature could change in the following 3 months, they may lose the opportunity to send warnings about the sudden change in a certain day.

6.4 Local polynomial regression and the tube formula

Wasserman suggested a method in building confidence bands for functions using local linear regression. The idea using a tube formula inspired the methods building simultaneous bands (Wasserman, 2006).

Suppose we have sample values (x_i, y_i), $i = 1, \ldots, n$. The linear regression model is

$$y_i = f(x_i) + \epsilon_i, i = 1, \ldots, n,$$

where $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2$. We want to estimate the function value f at x. Let p be the degree of polynomial which is used later (Fan, 1992). The weight function at x_i is

$$w_i(x) = K\left(\frac{x_i - x}{h}\right),$$

where h is the bandwidth and the kernel K is the tricube kernel

$$K(t) = \frac{70}{81}(1 - |t|^3)^3 I(t),$$

where

$$I(t) = \begin{cases} 1 & \text{if } |t| \leq 1; \\ 0 & \text{if } |t| > 1. \end{cases}$$

Define the polynomial matrix

$$X_x = \begin{pmatrix} 1 & x_1 - x & \cdots & \frac{(x_1-x)^p}{p!} \\ 1 & x_1 - x & \cdots & \frac{(x_2-x)^p}{p!} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x & \cdots & \frac{(x_n-x)^p}{p!} \end{pmatrix},$$

and W_x be the $n \times n$ diagonal matrix with ith diagonal component $w_i(x)$. The estimate is given by the following linear combination

$$\hat{f}_n(x) = \sum_{i=1}^{n} l_i(x) y_i,$$

where $l(x)^T = (l_1(x), \ldots, l_n(x))$ and

$$l(x)^T = e_1^t (X_x^T W_x X_x)^{-1} X_x^T W_x,$$

for $e_1 = (1, 0, \ldots, 0)^T$.

The choice of the bandwidth h could be obtained from cross-validation on the minimization of loss function. On one side, it is a proper way to get an optimal h. On another side, the cross-validation process takes more computing time. The leave-one-out cross-validation score is

$$cv(h) = \hat{R}(h) = \sum_{i=1}^{n} (y_i - \hat{f}_{-i}(x_i))^2,$$

where $\hat{f}_{-i}(x_i)$ is the estimator obtained by omitting the ith pair (x_i, y_i).

The next step is to estimate the variance σ^2. We noticed that σ^2 increases with x which means that the data are heteroscedastic. So the nonconstant variance is considered. We assume

$$y_i = f(x_i) + \sigma(x_i)\epsilon_i.$$

At first, we estimate $f(x)$ with $\hat{f}_n(x)$ provided before. Then define $Z_i = \log(y_i - \hat{f}_n(x_i))^2$. We regress z_i's on x_i's with the same nonparametric method to get an estimate $\hat{q}(x)$ of $\log \sigma^2(x)$. So we have an estimate

$$\hat{\sigma}^2(x) = e^{\hat{q}(x)}.$$

Then we consider building the confidence bands. An approximate confidence band for $f(x)$ is

$$B(x) = \hat{f}_n(x) \pm c\widehat{\sigma}(x)||l(x)||,$$

for $c \geq 0$ and $a \leq x \leq b$ where $||\cdot||$ is a L2 norm. We suppose that σ is known and have the following formula

$$P\Big(E(\hat{f}_n(x)) \notin \text{B(x) for some } x \in [a,b]\Big) = P\Big(max_{x\in[a,b]} \frac{|\hat{f}_n(x) - E(\hat{f}_n(x))|}{\sigma||l(x)||} > c\Big)$$
$$= P\Big(max_{x\in[a,b]} \frac{|\sum_i \epsilon_i l_i(x)|}{\sigma||l(x)||} > c\Big)$$
$$= P\Big(max_{x\in[a,b]} |W(x)| > c\Big),$$

where $W(x) = \sum_{i=1}^{n} Z_i T_i(x)$, $Z_i = \frac{\epsilon}{\sigma} \sim N(0,1)$ and $T_i(x) = \frac{l_i(x)}{||l(x)||}$. The $W(x)$ is a Guassian process.

The probability is shown by a tube formula (Sun and Loader, 1994)

$$P\Big(max_x |\sum_{i=1}^{n} Z_i T_i(x)| > c\Big) \approx 2(1 - \Phi(c)) + \frac{\kappa_0}{\pi} e^{-\frac{c^2}{2}},$$

where c is given by the following formula

$$2(1 - \Phi(c)) + \frac{\kappa_0}{\pi} e^{-\frac{c^2}{2}} = \alpha, \tag{6.1}$$

and $\Phi(\cdot)$ is a cdf of a standard normal distribution and

$$\kappa_0 = \int_a^b ||T'(u)|| du,$$

where $T(u) = (T_1(u), ..., T_n(u))$ with $T_i(u) = \frac{l_i(u)}{||l(u)||}$ and $T'(u) = (T_1'(u), ..., T_n'(u))$ with $T_i'(u) = \frac{\partial T_i(u)}{\partial u}$.

The value of κ_0 could be approximated by

$$\kappa_0 = \sum_{i=1}^{s} \int_{v_{i-1}}^{v_i} ||T'(u)|| du \approx \sum_{i=1}^{s} ||T(v_i) - T(v_{i-1})||.$$

If $[a,b]$ could be partitioned into $a = v_0 < \ldots < v_s = b$, then the L2 norm could be obtained by the "Composite Simpson's 3/8 rule" (Matthews, 2008).

6.5 Methods for simultaneous bands

From our literature review chapter, we summarized the existing ways to build simultaneous bands. The tube method is the key in the development of these methods (Wasserman, 2006). In general, there are two ways to estimate c from (6.1): Monte Carlo Quantile methods and Quasi Monte Carlo methods.

The Monte Carlo Quantile methods are resampling methods that obtain Monte Carlo draws from a multivariate (MVT) normal distribution with mean estiamtor and the variance-covariance matrix. Proper quantiles are calculated from the draws to obtain the lower and upper bounds. A typical method is Supt method as in Montiel Olea and Plagborg-Møller (2019).

Quasi Monte Carlo methods are under the assumption that the simulated draws are from the MVT normal distribution and the confidence bands can then be generated to approximate the desired probability. A typical method is the optimization method which we will discuss later in Robertson et al. (2020).

6.6 Component-wise simultaneous confidence bands

Now we introduce a new way to build simultaneous confidence bands instead of confidence intervals to achieve a better coverage for the whole density. To construct point-wise interval estimates for the kernel estimators via the batch means procedure is quite straightforward. But the proportion of confidence bands which could cover the density curve of f approaches 0 as m increases. It is also not appropriate to use the multiplicity corrections. For example, Bonferonni bands become wider as m increases.

Our goal is to develop functional confidence bands with the entire true density f within the bands given a level of confidence. From our previous discussion, we introduce the true multivariate nature of estimation. We assume a Markov CLT by the Monte Carlo error, $\hat{f}_n(\overrightarrow{w}) - f(\overrightarrow{w})$. There is a $m \times m$ positive definite matrix Σ satisfying

$$\sqrt{n}\left(\hat{f}_n(\overrightarrow{w}) - f(\overrightarrow{w})\right) \xrightarrow{\mathrm{d}} N_m(0, \Sigma), \tag{6.2}$$

when $n \to \infty$. The estimation of Σ is necessary in the application of CLT in (5.2). Although the computing procedure is fixed with even moderate m or n. Since the calculations of the batch type estimators in our work are more than 200 times faster in the similar dimension problems, our estimators are more appropriate (Liu and Flegal, 2018).

Denote $n = a_n b_n$, where a_n is the number of batches and b_n is the corresponding batch size. Define estimator $\tilde{f}_k(w)$ for the batch k at a random point w. A batch mean estimator of Σ is

$$\hat{\Sigma} = \frac{b_n}{a_n - 1} \sum_{k=0}^{a_n - 1} \left(\tilde{f}_k(\overrightarrow{w}) - \hat{f}_n(\overrightarrow{w})\right)\left(\tilde{f}_k(\overrightarrow{w}) - \hat{f}_n(\overrightarrow{w})\right)^T. \tag{6.3}$$

We want to build simultaneous $1 - \alpha$ confidence bands for f which means that we want to find a bunch of lower bounds L_i and upper bounds U_i for $i = 1, \ldots, m$ such that

$$P\Big(f(w_i) \in [L_i, U_i] \text{ for every i=1,...,m } \Big) \approx 1 - \alpha.$$

We proposed a quantile-based method in building simultaneous confidence bands. The bands were proved equal to the supt method (which will be discussed in the next section) asymptotically. The details are in the appendix section.

Algorithm 1. Component-wise simultaneous method

1: Compute the variance-covariance estimate $\hat{\Sigma}$.

2: Draw a_n i.i.d normal vectors $\hat{V}^{(l)} \sim N_m(0_m, \hat{\Sigma})$, $l = 1, \ldots, a_n$.

3: Samples $T^{(l)} = (\hat{f}_n(w_1) + c^{(l)}\hat{\sigma}_1, \ldots, \hat{f}_n(w_m) + c^{(l)}\hat{\sigma}_m)$ where $c = \hat{V}_k^{(l)}{\hat{\sigma_k}}^{-1}$ and $k = argmax|\hat{V}_j^{(l)}{\hat{\sigma_j}}^{-1}|$ for each l.

4: $\hat{C} = \mathsf{X}_{j=1}^{m}[T_{j,\alpha/2}, T_{j,1-\alpha/2}]$ where $T_{j,\tau}$ is the empirical τ quantile of T_j, $j = 1, \ldots, m$.

6.7 Supt simultaneous confidence bands

For any given $c > 0$, the confidence band is

$$\hat{B}(c) = [\hat{f}_n(w_1) - \hat{\sigma}_1 c, \hat{f}_n(w_1) + \hat{\sigma}_1 c] \times [\hat{f}_n(w_2) - \hat{\sigma}_2 c, \hat{f}_n(w_2) + \hat{\sigma}_2 c] \times \cdots \times [\hat{f}_n(w_m) - \hat{\sigma}_m c, \hat{f}_n(w_m) + \hat{\sigma}_m c].$$

For the true function vector $f(\overrightarrow{w}) = \Big(f(w_1), \ldots, f(w_m)\Big)$,

$$P\big(f(\overrightarrow{w}) \in \hat{B}(c)\big) \to P\big(\max_{j=1,\ldots,m} |\Sigma_{jj}^{-1/2} V_j| \leq c\big),$$

where $V \sim (V_1, \ldots, V_m)' \sim N_m(0_m, \Sigma)$ and Σ_{jj} is the jth diagonal element of Σ. Define the ζ-quantile of the previous random variable as a function of the variance-covariance matrix Σ:

$$q_\zeta(\Sigma) = Q_\zeta\big(\max_{j=1,\ldots,m} |\Sigma_{jj}^{-1/2} V_j|\big). \tag{6.4}$$

We obtain the sup-t band by choosing $c = q_{1-\alpha}(\Sigma)$ (the sup-t critical value), yielding a simultaneous coverage probability of precisely $1 - \alpha$. The algorithm for Supt band is as follows.

Algorithm 2. Supt simultaneous band

1: Compute the variance-covariance estimate $\hat{\Sigma}$.

2: Draw W i.i.d normal vectors $\hat{V}^{(l)} \sim N_m(0_m, \hat{\Sigma})$, $l = 1, \ldots, W$.

3: Define $\hat{q}_{1-\alpha}$ as the empirical $1-\alpha$ quantile of $\max_j |\hat{\Sigma}_{jj}^{-1/2} \hat{V}_j^{(l)}|$ across $l = 1, \ldots, W$.

4: $\hat{C} = \hat{B}(\hat{q}_{1-\alpha}) = \mathsf{X}_{j=1}^{m} [\hat{f}_n(w_j) - \hat{\sigma}_j \hat{q}_{1-\alpha}, \hat{f}_n(w_j) + \hat{\sigma}_j \hat{q}_{1-\alpha}]$.

The algorithms above provide a nice setup to validate our methodology since it has solid theories and well-defined metrics. So the next step is to use some simulations to check whether our component-wise confidence bands and the Supt bands are identical or not.

6.8 Optimization method

A Quasi Monte Carlo method to build simultaneous confidence bands is to find hyperrectangular regions between pointwise bands and Bonferroni bands (Robertson et al., 2020). Since this is a method that optimizes the confidence region, we call it an "optimization" method.

The approach is to consider hyperrectangular regions $C_{LB} \subseteq C_{UB}$ where C_{LB} has coverage no greater than $1-\alpha$ while C_{UB} has coverage at least $1-\alpha$. There exists some hyperrectangular region, C_α, between these, $C_{LB} \subseteq C_\alpha \subseteq C_{UB}$, which will have coverage $1-\alpha$.

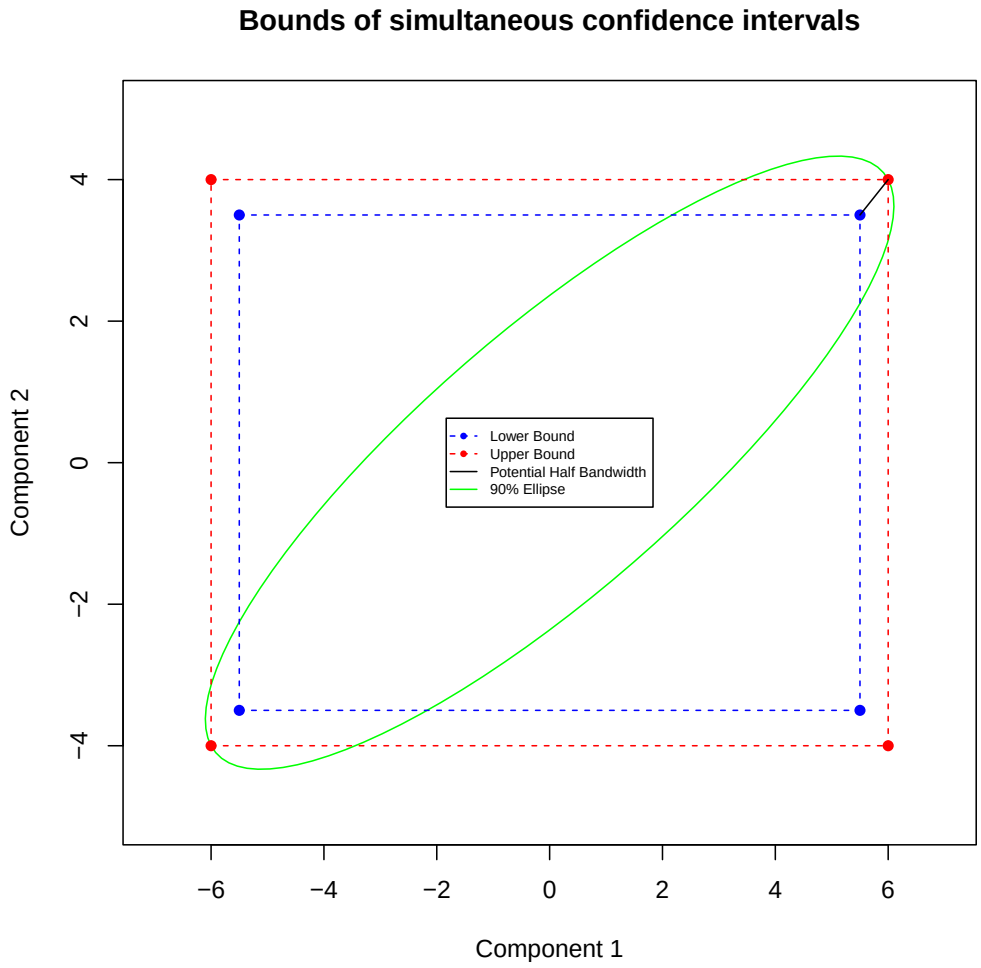


Figure 6.1: Simultaneous confidence interval visualization for the optimization method.

For $z > 0$, consider the hyperrectangular confidence regions of the form

$$C_{SI}(z) = \mathsf{X}_{i=1}^{m} \left[\hat{f}_n(w_i) - z\hat{\sigma}_i, \hat{f}_n(w_i) + z\hat{\sigma}_i \right],$$

Setting $z = z_{\frac{\alpha}{2}}$ gives pointwise (uncorrected) intervals that simultaneously have coverage no greater than $1-\alpha$ as $C_{LB} := C_{SI}(z_{\frac{\alpha}{2}})$. With $z = z_{\frac{\alpha}{2m}}$, we have the Bonferroni-corrected hyperrectangular region that has coverage at least $1-\alpha$ and set $C_{UB} := C_{SI}(z_{\frac{\alpha}{2m}})$.

The idea is to find z^* such that $z_{\frac{\alpha}{2}} \leq z^* \leq z_{\frac{\alpha}{2m}}$ and $C_{SI}(z^*)$ has coverage $1-\alpha$. We start simulating $U \sim N_m(\hat{f}_n(\overrightarrow{w}), \hat{\Sigma})$. It is easy to see $P(U \in C_{LB}) \leq 1-\alpha$ and $P(U \in C_{UB}) \geq 1-\alpha$. As $Pr(U \in C_{SI}(z))$ is strictly increasing as z increases, we then use the bisection method between $z_{\frac{\alpha}{2}}$ and $z_{\frac{\alpha}{2m}}$ to find z^* such that $Pr(U \in C_{SI}(z^*)) \approx 1-\alpha$.

Algorithm 3. Optimization method

1: Compute the variance-covariance estimate $\hat{\Sigma}$.

2: Draw W i.i.d normal vectors $\hat{V}^{(l)} \sim N_m(\hat{f}_n(\overrightarrow{w}), \hat{\Sigma})$, $l = 1, \ldots, W$.

3: Find z^* in the interval $[z_{\frac{\alpha}{2}}, z_{\frac{\alpha}{2m}}]$ such that $Pr(U \in C_{SI}(z^*)) \approx 1-\alpha$.

4: $\hat{C} = \mathsf{X}_{i=1}^{m}[\hat{f}_n(w_i) - z^*\hat{\sigma}_i, \hat{f}_n(w_i) + z^*\hat{\sigma}_i]$.

This method combined the properties of pointwise and Bonferroni approaches to find confidence regions with a confidence level $1-\alpha$. With certain assumptions, the confidence level is specified and guaranteed. The drawback about the running time for applying this method is huge if there are many points to be estimated. More examples will be given in the following chapters.

6.9 Confidence band comparison metrics

We compare various confidence bands using a number of metrics in a variety of simulation examples and real data analysis. We consider confidence bands (L_i, U_i) for $i = 1, \ldots, m$. If the simulation is repeated r times and the bands cover the true function a out of r times simultaneously, we call the coverage probability is $C_{prob} = \frac{a}{r}$.

Monte Carlo standard error (MCSE) is an estimate of the inaccuracy of Monte Carlo samples. If the confidence level is $1 - \alpha$, the Monte Carlo standard error for a confidence band with coverage probability C_{prob} is

$$\sqrt{\frac{(1 - C_{prob})C_{prob}}{r}}.$$

If the desired coverage probability $1 - \alpha$ is within $1.96 \times MCSE$ (with 90% confidence) from the coverage probability C_{prob}, the average coverage is regarded as reasonable.

For example, if confidence level is $1 - \alpha = 0.9$ and the simulation time $r = 200$, the Monte Carlo standard error for a coverage probability $C_{prob} = 0.92$ is 0.019. Since the desired coverage 90% is in the confidence interval

$$[0.92 - 1.96 \times 0.019, 0.92 + 1.96 \times 0.019] = [0.882, 0.958].$$

So this confidence band has reached the desired coverage 90% with certainty accuracy.

The average width for confidence bands (L_i, U_i) is the averaged width along all the points

$$\frac{1}{m}\sum_{i=1}^{m}(U_i - L_i).$$

If we set the pointwise method width $(U_p(w_i) - L_p(w_i))$ as reference. Bonferroni confidence bands are $(L_b(w_i), U_b(w_i))$. The average of ratios from Bonferroni bands' widths to pointwise bands' widths at each point is called the relative width for Bonferroni methods

$$\frac{1}{m}\sum_{i=1}^{m}\frac{U_b(w_i) - L_b(w_i)}{U_p(w_i) - L_p(w_i)}.$$

6.10 Correlated points simulation procedure

In the real world, data can possibly be correlated. Our methodology also targets at building confidence bands for dependent dataset. With high demands and needs for motivating the real data experiments, we need to simulate correlated $x's$. In the future simulation study, we want our functional observations to be correlated. However, from the level of function values, it is difficult to achieve this. So with the help of Metropolis-Hastings algorithm listed below, we could simulated the correlated $x's$ and then generate the function values with these $x's$. Then the function values are correlated. Since we want to simulated $x's$, the $x's$ should be correlated from nature of MCMC methods. We will apply the random walk chains method to simulate the distribution $Uniform(a, b)$. In the statistical fields where the equilibrium (invariant, stationary) distribution is a posterior distribution such as Bayesian inference, there is a great need for MCMC methods.

Suppose we want to simulate draws from denstiy function $f(\cdot)$. We need to construct a suitable Markov chain with f as its stationary distribution. A critical problem is

that we could only observe serially dependent t observations from $\{X_t\}$. The algorithm is given below.

Algorithm 4 Metropolis-Hastings algorithm

1: Setting $X_0 = x_0$.

2: For $t = 1, \ldots, n$, sample a candidate value $X^* \sim g(\cdot|x_{t-1})$ where g is the proposal distribution.

3: Compute the MH ratio $R(x_{t-1}, X^*)$

$$R(x_{t-1}, X^*) = \frac{f(x^*)g(x_{t-1}|x^*)}{f(x_{t-1})g(x^*|x_{t-1})}.$$

4: Set

$$X_t = \begin{cases} x^* & \text{w.p } \min\{R(x_{t-1}, X^*)\}; \\ x_{t-1} & \text{otherwise.} \end{cases}$$

In our case, the function $f(\cdot)$ is $Uniform(a, b)$. We generate X^* such that $\epsilon \sim h(\cdot)$ and set $X^* = X_{t-1} + \epsilon$, then $g(x^*|x_{t-1}) = h(x^* - x_{t-1})$. The common choices of $h(\cdot)$ are symmetric mean 0 with a scale parameter σ, for example $Uniform(-\sigma, \sigma)$ and $N(0, \sigma)$. We use $h(\cdot) = N(0, \sigma)$ with a reasonable σ.

With such $h(\cdot)$, the MH ratio is

$$R(x_{t-1}, X^*) = \frac{f(x^*)}{f(x_{t-1})} = I(a \leq x^* \leq b)I(a \leq x_{t-1} \leq b).$$

So the algorithm is specified as follows.

Algorithm 5 Random walk chain specified

1: Let $x_0 = \frac{a+b}{2}$.

2: For $t = 1, \ldots, n$, a candidate $x^* = x_{t-1} + \epsilon$, where $\epsilon \sim N(0, \sigma)$.

3: Simulate $u \sim Uniform(0, 1)$. If $u < R(x_{t-1}, X^*)$, $x_t = x^*$. Else $x_t = x_{t-1}$.

This procedure of generating $x's$ is well-defined and could provide correlated draws. With the correlated $x's$, we could generate the corresponding functional values.

6.11 Simulation example

In order to get a sense of how our method could be applied in building confidence bands for functions, we will look at the simulations for some smooth functions. Consider the target function

$$f(x) = 0.3\exp(x) + 2\sin(x) - 2x,$$

where $x's$ are simulated from $Uniform(-1, 1)$ (using random walk to expand correlations to the draws). We generate the sample function values y_i's by $y_i = f(x_i) + \epsilon_i$ with the noise $\epsilon_i \sim N(0, 0.5^2)$. The number of points to be estimated is $m \in \{10, 20, 40, 80\}$ and the points are equally spaced between $(-0.8, 0.8)$. The bandwidth parameter for the kernel $K_\lambda(x_0, x)$ is $\lambda = 0.15$ and confidence level is $1 - \alpha = 90\%$. The simulation is replicated for 200 times. Tables 6.1-6.3 and Figure 6.4 show the simulation results.

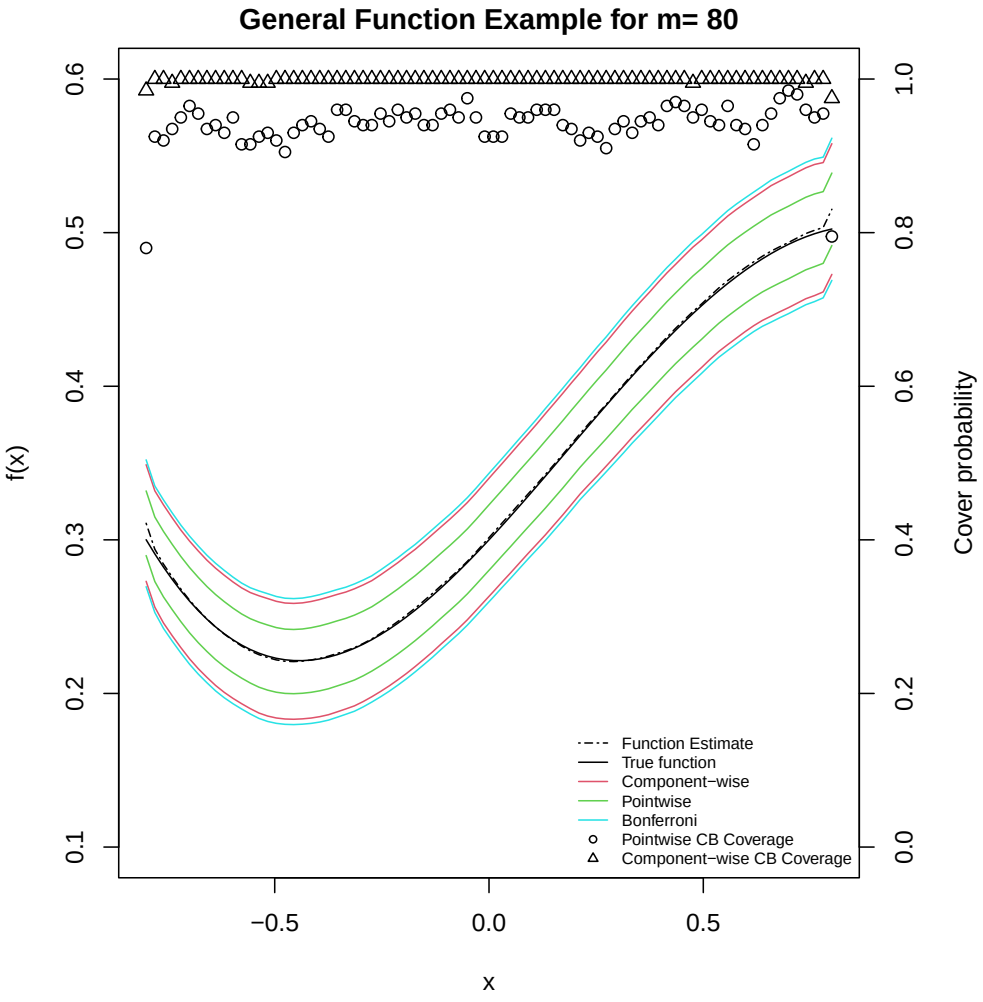


Figure 6.2: Example for comparing different confidence bands.

Table 6.1: General function example: coverage probability comparisons among different methods.

m	10	20	40	80
Pointwise	0.335	0.255	0.16	0.13
Bonferroni	0.92	0.93	0.955	0.97
Component-wise	0.885	0.9	0.94	0.94
Optimization	0.92	0.92	0.945	0.95
Supt	0.89	0.915	0.94	0.925

Table 6.2: General function example: average widths of bands comparisons among different methods.

m	10	20	40	80
Pointwise	0.044	0.044	0.044	0.044
Bonferroni	0.069	0.075	0.081	0.086
Component-wise	0.067	0.073	0.077	0.079
Optimization	0.068	0.074	0.078	0.08
Supt	0.068	0.073	0.077	0.079

Table 6.3: General function example: relative widths of bands comparisons among different methods.

m	10	20	40	80
Pointwise	1	1	1	1
Bonferroni	1.566	1.707	1.838	1.962
Component-wise	1.532	1.666	1.759	1.804
Optimization	1.557	1.682	1.775	1.823
Supt	1.546	1.674	1.765	1.807

From Table 6.1, when number of grid points m is small as 10 or 20, the Bonferroni method coverage is reasonable as the simultaneous methods. The confidence interval for Bonferroni method coverage probability 0.93 when $m = 20$ is $[0.895, 0.965]$ where the desired coverage is near the interval boundary. As m increases to 40, Bonferroni and simultaneous

methods depart from the desired coverage probability of 90%. The coverage probabilities for simultaneous are also conservative probably due to an improper bandwidth choice λ.

When $m = 80$, the Bonferroni method becomes too conservative to provide the right coverage with coverage probability confidence interval $[0.946, 0.994]$, which is extremely far from the desired probability 90%. The simultaneous method like supt method (coverage probability: 0.925) can still capture the right coverage with MCSE 0.019 and confidence interval $[0.888, 0.962]$. Overall, as m increases, the pointwise method coverage becomes smaller from the lack of multiplicity correction and fails to provide the desired coverage.

From Tables 6.2 and 6.3, the Bonferroni bands become wider relatively as m increase. For example, when $m = 10$, Bonferroni band is 1.2% wider than Supt band. When $m = 80$, Bonferroni band is 8.6% wider than Supt band.

Overall we inferred that the simultaneous bands become relatively narrower compared to Bonferroni bands as the number of points m increase. When the number of grid points m is large, the Bonferroni bands are too conservative to use. As expected, pointwise bands continue to lose the integrity in covering the whole function as m increases. We will deliver more examples to compare the confidence bands later.

6.12 Pros and cons for Wasserman's method

In Section 6.4, Wasserman provided a way in building confidence intervals for functional estimates. This methods provide a well-defined procedure for estimating functions and giving confidence bands. The techniques in the procedure are fully developed with solid theories. If we aim to build confidence bands for functions that do not require predictions

on the unknown points, Wasserman's method is ready to go. The most important contribution of Wasserman's method is about the tube formula which is the fundamental of the simultaneous bands.

Here are some drawbacks for this approach. At first, the approach could only estimate the function value $f(x_0)$ when there are some observations at x_0. When it comes to predictions for some arbitrary point x^* where there is no such y observed at x^*, the approach will be out of use. Our proposed method could provide estimates and build confidence bands for any point x^* even we do not have observations at x^*.

Furthermore, the computing load for Wasserman's method is high. Through the whole methodology, the cross-validation for choosing h, estimating κ and finding the root of equation (6.1) are the time consumed. Since these steps could not be avoided, the running time for the procedure is long. With some simulations, our proposed method could obtain a shorted runtime and hence achieved a computing efficiency. Another drawback for Wasserman's method is about the estimation for the L2 norm of a function. Its approximation accuracy varies. The performance of estimation and coverage will be affected by the way of approximation. Overall our methodology outperformed the Wasserman's method in the three aspects above.

6.12.1 Example

To illustrate the differences among different methods, we use the data from 5.59 Example (LIDAR) (Wasserman, 2006). The data is heteroscedastic with 221 paired (x_i, y_i)'s. We use the suggested $h = 37$ and $p = 1$ for their method.

We denote the true function $f(x)$ by the Nadaraya–Watson kernel estimator

$$f(x) = \sum_{i=1}^{n} L_i(x) y_i,$$

where

$$l_i(x) = \frac{K(\frac{x-x_i}{h})}{\sum_{i=1}^{n} K(\frac{x-x_i}{h})}.$$

The kernel K is defined before and $h = 20$. We simulate from this function using

$$y_i = f(x_i) + e(x_i),$$

where the noise is

$$e(u) \sim \begin{cases} Uniform(-0.03, 0.03) & \text{if } u < 450; \\ Uniform(-0.08, 0.08) & \text{if } 450 \leq u < 600; \\ Uniform(-0.2, 0.2) & \text{others.} \end{cases}$$

The $x's$ are simulated from $Uniform(380, 730)$ (using random walk from Algorithm 5 to get correlated samples). The number of points to be estimated is $m = 10$ and the points are equally spaced between $(480, 630)$. The bandwidth parameter for the kernel $K_\lambda(x_0, x)$ is $\lambda = 0.2$. The replicates are 100 and the confidence level is 80%. The parameters we used are $n = 20,000$, the number of batches $a_n = 200$ and the batch size $b_n = 100$.

From Table 6.4, the local polynomial regression procedure proposed by Wasserman has the coverage 0 for different m choices so this procedure could not reach the desired

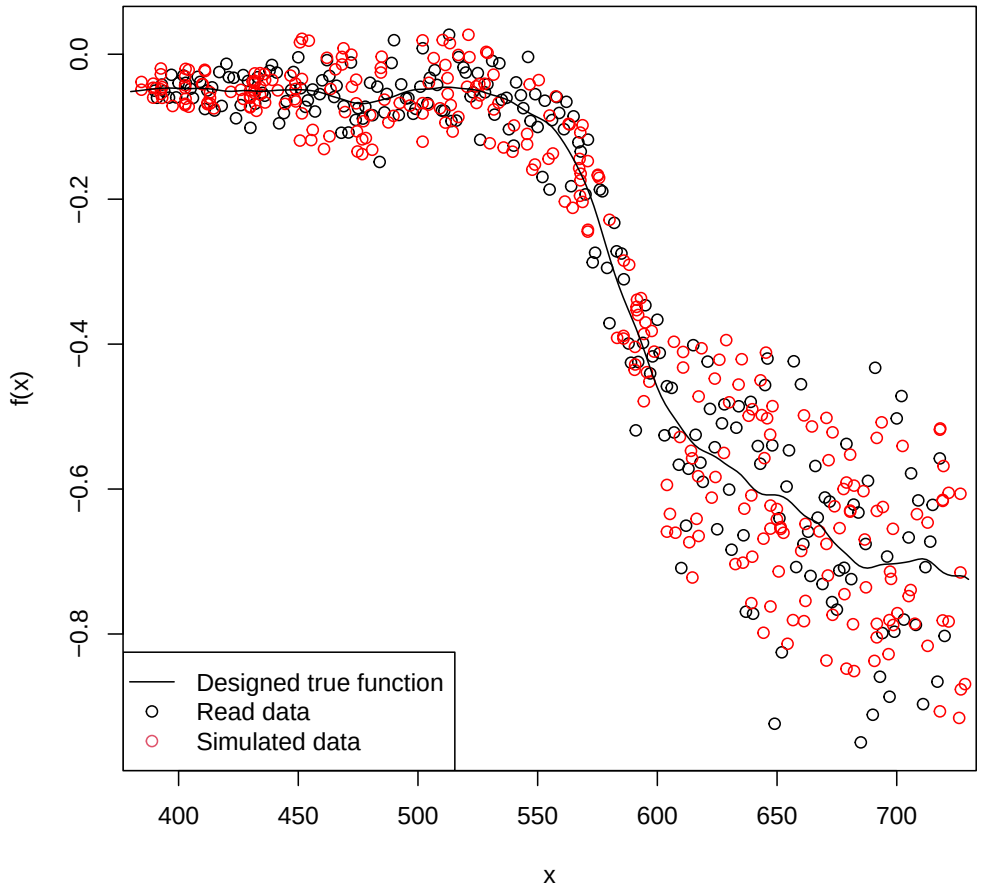


Figure 6.3: Wasserman's example: simulated data and the real data.

Table 6.4: Total coverage probability comparisons for Wasserman's method with 80% confidence level

m	10	40
Wasserman	0	0
Pointwise	0.44	0.29
Bonferroni	1	1
Component-wise	1	0.97

probability 80%. While the Bonferroni and Component-wise methods are conservative to use, we will investigate more with examples. Though the estimation procedure from Wasserman provided theoretical support for the simultaneous bands, we would keep using our functional estimators described in the general function estimation chapter.

We could tell that our simultaneous methods have higher coverage probabilities over Wasserman's method with accurate function estimators and wider band widths.

6.13 Appendix: asymptotic equivalence between component-wise and Supt methods

To show the asymptotic equivalence of component-wise (algorithm 1) and Supt (algorithm 2) methods, we will have several settings. In algorithm 2, we denote

$$d = \max_j |\hat{\Sigma}_{jj}^{-1/2} \hat{V}_j^{(l)}|,$$

across $l = 1, \ldots, N$ in step 3. In step 4, the upper and lower bound is

$$(\hat{f}_n(w_j) - \hat{\sigma}_j d_{1-\alpha}^{(l)}, \hat{f}_n(w_j) + \hat{\sigma}_j d_{1-\alpha}^{(l)}),$$

for each j. In algorithm 1, we have

$$(T_{j,\alpha/2}, T_{j,1-\alpha/2}) = (\hat{f}_n(w_j) + c_{\alpha/2}^{(l)} \hat{\sigma}_j, \hat{f}_n(w_j) + c_{1-\alpha/2}^{(l)} \hat{\sigma}_j),$$

for arbitrary j. It is obvious that $c^{(l)}$ has a normal distribution with mean zero.

Suppose z is a random variable having a normal distribution $N(0, \sigma)$ and $z^* = |z|$ as another variable with the absolute value. For a certain $100(1 - \alpha/2)$th percentile for z, $z_{1-\alpha/2}$,it is easy to prove

$$z_{1-\alpha/2} = |z|_{1-\alpha}.$$

Similarly,

$$z_{\alpha/2} = -|z|_{1-\alpha}.$$

If N is large enough, for the absolute values $d^{(l)} = |c^{(l)}|$ should have the following properties

$$c^{(l)}_{\alpha/2} = -d^{(l)}_{1-\alpha},$$

and

$$c^{(l)}_{1-\alpha/2} = d^{(l)}_{1-\alpha}.$$

Thus the asymptotic bounds are equal for these two algorithms.

For visualization using the following two-component mixed normal distribution function

$$0.5N(-1, (2/3)^2) + 0.5N(1, (2/3)^2), \tag{6.5}$$

The number of points to be estimated is $m = 20$ grid point equally spaced along $(-2, 2)$ from different methods with confidence level 90%. In each simulation, $n = 400,000$ draws were generated. With the number of batches $a_n = 400$ and the batch size $b_n = 1000$. The

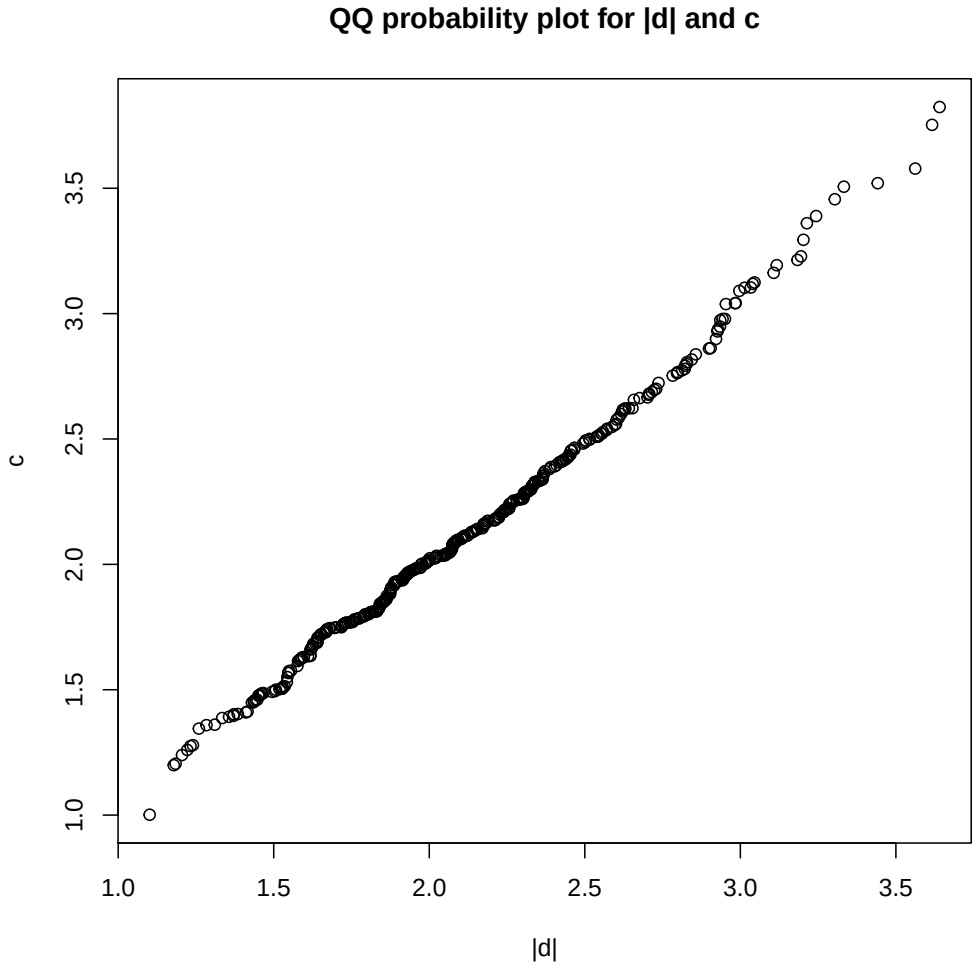


Figure 6.4: Visualization for the asymptotic equivalence between component-wise and Supt methods.

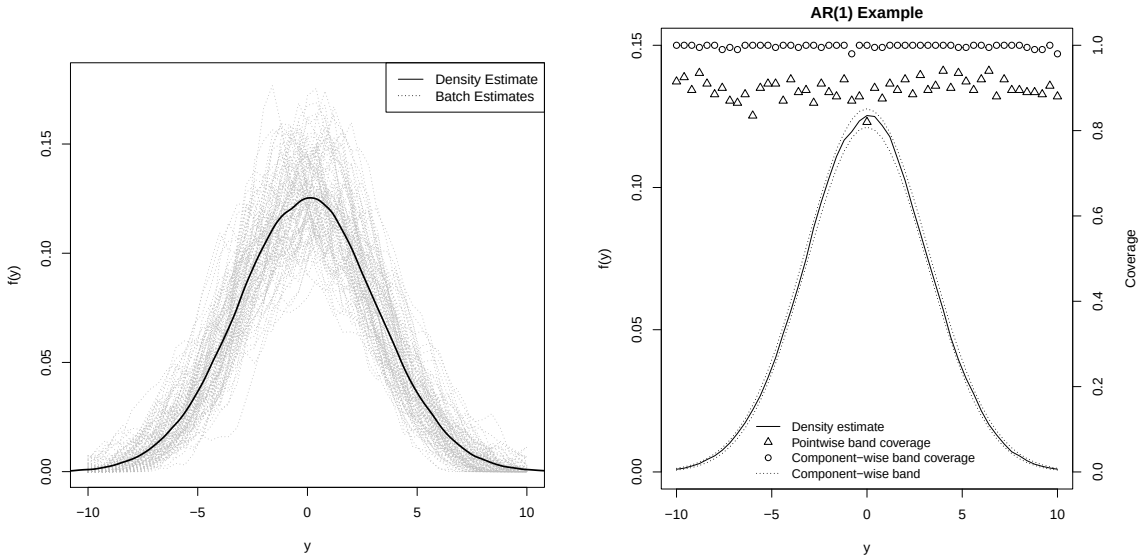
number of draws for $d^{(l)}$ and $|c^{(l)}|$ is 400. The Quantile-Quantile (QQ) probability plot for $d^{(l)}$ and $|c^{(l)}|$ is shown in Plot 6.4.

Chapter 7

Simulation Examples

To explore the differences in the pointwise, Bonferroni and simultaneous confidence bands, several simulation experiments were conducted. With proper parameters and solid assumptions, our method achieved a good overall coverage behavior with more accurate results compared to pointwise and Bonferroni approaches.

For simulation studies, to create the dependent functional values $f(x)$, a practical way is to create a correlated vector of $x's$. With the Metropolis-Hastings algorithm (Metropolis et al., 1953), we use the random walk chain method to achieve this aim. That is, the simulated draws from the procedure, $x's$, are correlated and hence so are the corresponding function values. Simulation studies include AR(1) model, mixed normal distribution for densities, and a general function example.



(a) Kernel density and first 100 batch estimates.

(b) Pointwise and simultaneous confidence bands.

Figure 7.1: AR(1) Process kernel density estimate with 90% confidence bands.

7.1 Density estimation example: AR(1) model

Consider an autoregressive process of order 1 or AR(1), i.e.

$$X_{i+1} = \phi X_i + \epsilon_i, \text{for } i = 1, 2, \ldots,$$

where ϵ_i are i.i.d $N(0, 1)$. This Markov chain is geometrically ergodic for $|\phi| < 1$. The estimated marginal density f of X is $N(0, \frac{1}{1-\phi^2})$. We aim to compare the performance of our procedure among different methods and the one with bias correction. Now we estimate f, $\phi = 0.95$ and $m = 51$ equally spaced skeleton points on $[-10, 10]$. The selected bandwidth is $h = 0.162$. Since we have already known the true marginal density f, we could calulate

the overall coverages for confidence bands. We aim to compare the behaviors of pointwise band, Bonferroni band and various simultaneous bands.

The result for a single simulation is given in Figure 7.1a. The solid black line is the kernel density estimate of f from $n = 400,000$ Markov chain draws and the thin grey lines are kernel density estimates of f from the first 100 batches ($a_n = 400$) of size $b_n = 1000$. The simulation error could be estimated through Σ in (5.2) from the variability in the batch estimates. Then the pointwise confidence intervals and component-wise confidence bands could be constructed with $(1 - \alpha) = 0.9$. These are shown in Figure 7.1b. The pointwise band had the desire coverage around 90% at each point while the component-wise band had higher coverage at each point which led to the simultaneous coverage over the whole density.

The simulation process is repeated 200 times for performance evaluation. Figure 7.1b plots y versus with observed pointwise coverage probability on the right axis with small open circles.

Table 7.1: AR(1) example: coverage probabilities and relative widths.

Method	Pointwise	Bonferroni	Component-wise	Optimization	Supt
Coverage	0.095	0.93	0.895	0.89	0.885
Relative Width	1	1.882	1.787	1.791	1.79

In Table 7.1, the confidence bands from pointwise bands result in 37 of 200 (9.5%) that contain the entire true known density. It is expected since there is no multiplicity correction for the pointwise bands. In the mean time, the component-wise bands contain the density function in 179 out of 200 simulations (89.5%) with $MCSE = 0.022$ and coverage confidence interval $[0.853, 0.937]$ which contained the desired 90% coverage. The Bonferroni

bands contain the density function in 186 out of 200 simulations (93%) with $MCSE = 0.022$ and coverage confidence interval $[0.895, 0.965]$ where 90% is included though. From the widths comparisons, the Bonferroni bands are conservative whose widths are 5.3% wider than Component-wise bands and 88.2% wider than the pointwise bands. All three simultaneous methods have similar relative widths.

7.2 Density estimation example: three-component mixed normal distribution

Consider the following three-component mixed normal distribution function

$$0.2N(0,(1)^2) + 0.2N(0.5,(2/3)^2) + 0.6N(13/12,(5/9)^2), \tag{7.1}$$

where $N(\mu, \sigma^2)$ is a normal distribution function with mean μ and standard deviation σ. In order to get correlated draws, the data were sampled from Metropolis–Hastings algorithm. The goal is to build the confidence bands for $m \in \{20, 40, 80, 120\}$ grid point equally spaced along $(-1, 1.5)$ from different methods with confidence level 90%.

In each simulation, $n = 400,000$ draws were generated. With the number of batches $a_n = 400$ and the batch size $b_n = 1000$, the density estimator $\hat{f}_n(\overrightarrow{w})$ and the batch estimators $\{\tilde{f}_k(\overrightarrow{w}), k = 1, \ldots, a_n\}$ were calculated. The optimal bandwidth is $h = 0.042$. The simulation was repeated for 200 replications.

Table 7.2: Three-component mixed normal density example: overall coverage probability comparisons among different methods.

m	20	40	80	120
Pointwise	0.08	0.03	0.01	0.01
Bonferroni	0.895	0.885	0.935	0.96
Component-wise	0.885	0.885	0.865	0.89
Optimization	0.88	0.875	0.89	0.91
Supt	0.88	0.875	0.88	0.91

Table 7.3: Three-component mixed normal density example: average widths of bands comparisons.

m	20	40	80	120
Pointwise	0.009	0.009	0.009	0.009
Bonferroni	0.015	0.016	0.017	0.018
Component-wise	0.015	0.016	0.016	0.017
Optimization	0.015	0.016	0.016	0.017
Supt	0.015	0.016	0.016	0.017

Table 7.4: Three-component mixed normal density example: relative widths of bands comparisons.

m	20	40	80	120
Pointwise	1	1	1	1
Bonferroni	1.707	1.838	1.962	2.031
Component-wise	1.692	1.81	1.868	1.887
Optimization	1.697	1.814	1.874	1.894
Supt	1.697	1.814	1.875	1.893

From Table 7.2, when the number of grid points m is not large (20 or 40), the Bonferroni method achieves the desired coverage 90% as do simultaneous bands. As m increases to 80, Bonferroni band starts showing conservation with coverage probability

0.935. This coverage probability has MCSE 0.017 and a confidence interval $[0.901, 0.969]$ which indicates that Bonferroni band is too conservative to achieve the 90% simultaneous coverage. The Component-wise band has the coverage probability 0.865 with MCSE 0.024 with confidence interval $[0.818, 0.912]$ which contains the desired probability. When $m = 120$, the Bonferroni band has a coverage 0.96 with MCSE 0.014 and a confidence interval $[0.933, 0.987]$ which is far away from the desired coverage 90%. For the pointwise band, the coverage probability decreases from 0.08 to 0.01 as m increases from 20 to 120 as expected.

From Tables 7.3 and 7.4, the Bonferroni band is 70.7% wider than pointwise band and only 0.6% wider than optimization band as a small $m = 20$. However, as m increases to 120, the Bonferroni band is 103.1% wider than pointwise band and nearly 7.2% wider than optimization band.

Overall, for different m choices, our simultaneous confidence bands have a consistent coverage probabilities near 90%. The Bonferroni band starts being conservative as m increases from 80.

7.3 General function example

For the function

$$f(x) = x + xsin(x),$$

where $x's$ are simulated from $Uniform(-2.5, 2.5)$ (using random walk to add correlations to the draws) and sample function values $y_i's$ are simulated through $y_i = f(x_i) + \epsilon_i$ with the noise $\epsilon_i \sim N(0, 0.5^2)$. The number of points to be estimated is $m \in \{40, 80, 120\}$ and the points are equally spaced between $(-2, 2)$. The bandwidth for the kernel $K_\lambda(x_0, x)$ is

$\lambda = 0.15$. In each simulation, $n = 20,000$ draws were generated. The number of batches $a_n = 100$ and the batch size $b_n = 200$. The confidence level is $1 - \alpha = 90\%$. The simulation was repeated for 200 replicates.

Table 7.5: General function $x + xsin(x)$ example: coverage probabilities of different confidence bands.

m	40	80	120
Pointwise	0.17	0.155	0.155
Bonferroni	0.89	0.935	0.95
Component-wise	0.84	0.87	0.9
Optimization	0.875	0.89	0.905
Supt	0.86	0.9	0.88

Table 7.6: General function $x + xsin(x)$ example: average widths of different confidence bands.

m	40	80	120
Pointwise	0.081	0.08	0.08
Bonferroni	0.149	0.158	0.163
Component-wise	0.143	0.146	0.146
Optimization	0.145	0.148	0.148
Supt	0.144	0.147	0.147

Table 7.7: General function $x + xsin(x)$ example: relative widths of different confidence different bands.

m	40	80	120
Pointwise	1	1	1
Bonferroni	1.838	1.962	2.031
Component-wise	1.781	1.814	1.826
Optimization	1.781	1.838	1.854
Supt	1.769	1.825	1.837

From Table 7.5, when $m = 40$, the confidence interval for component-wise coverage is $[0.789, 0.891]$ with MCSE 0.026. The confidence interval for optimization and Supt coverage are $[0.829, 0.921]$ and $[0.812, 0.908]$ which are more reasonable than the component-wise one. One thing to notice is that Bonferroni band achieved 0.89 which is the band nearest to 90% among all the methods. When $m = 80$, the Bonferroni band has the coverage 0.935 with a confidence interval $[0.901, 0.969]$ which starts showing the drawbacks of the over-correction for the multiplicity. When $m = 120$, the Bonferroni band continues gaining higher coverage as 0.95 where the confidence interval is $[0.920, 0.980]$ which is far away from the desired coverage 90%. For different m choices, optimization band has the most consistent coverage among three simultaneous bands.

In Tables 7.6 and 7.7, when $m = 20$, the Bonferroni band is only 3.2% wider than the optimization band. When $m = 120$, the Bonferroni band is almost 9.5% wider than the optimization band.

Overall, all three simultaneous bands meet the coverage requirements while the optimization band has the most consistent performance among them.

Chapter 8

Real Data Examples

Besides the simulation study, the real data examples need more restrictions. The real data should not be in comparative large scale without too much assumptions. Dominitz and Manski (2011) discussed a long-term survey on how residents thought economic status, which was called "Michigan's survey". Some variables in the dataset are worth exploring with our proposed methodology. Wasserman (2006) used the data LIDAR to illustrate the method in building confidence bands for the function. One drawback is that the unprediction of the points where no existing functional values are available. With multivariate setting, Wu (1998) stated the confidence intervals building in a time-varying model.

For the first real data case, responses for the survey in the percent that one thousand dollar investment will increase in value in the year ahead by each respondent are worth exploring from the Michigan survey data (Dominitz and Manski, 2011). We applied our methodology to build the simultaneous confidence bands of the percentages for 5 months and the results are within the scope.

In the second real data case, we built confidence bands for a variable in the MAGIC Gamma Telescope Data Set. We compared our simultaneous confidence bands with the uniform bands which have the same width for all the grid points (Dua and Graff, 2017; Cheng and Chen, 2019).

The third data case is for the time-varying model by measuring the uncertainties for different model coefficients (Wu et al., 1998). We compared our methodology with the pointwise intervals and the Bonferroni method.

The last data case is the LCD projector data to illustrate our simultaneous confidence bands in a Bayesian reliability model setting (Hamada et al., 2008).

8.1 Real data example: Michigan's survey

We could build confidence bands from a survey dataset (Dominitz and Manski, 2011). From June 2002 to August 2004 (27 months), surveys were given out in Michigan for each month to consumers approximately 500 adult men and women from coterminous area. Those surveys were completed by telephone and has a rotating panel basis design. In this design, the majority of individuals (approximately 300) responds at the first time and the remaining people (approximately 200) are those who were interviewed half year earlier (Curtin, 1982).

From the variables provided in the data, it is of great interest to analyze how the people think about their investments. The variable is defined by the the percent that one thousand dollar investment will increase in value in the year ahead by each respondent. The continuous value is from 0 to 100.

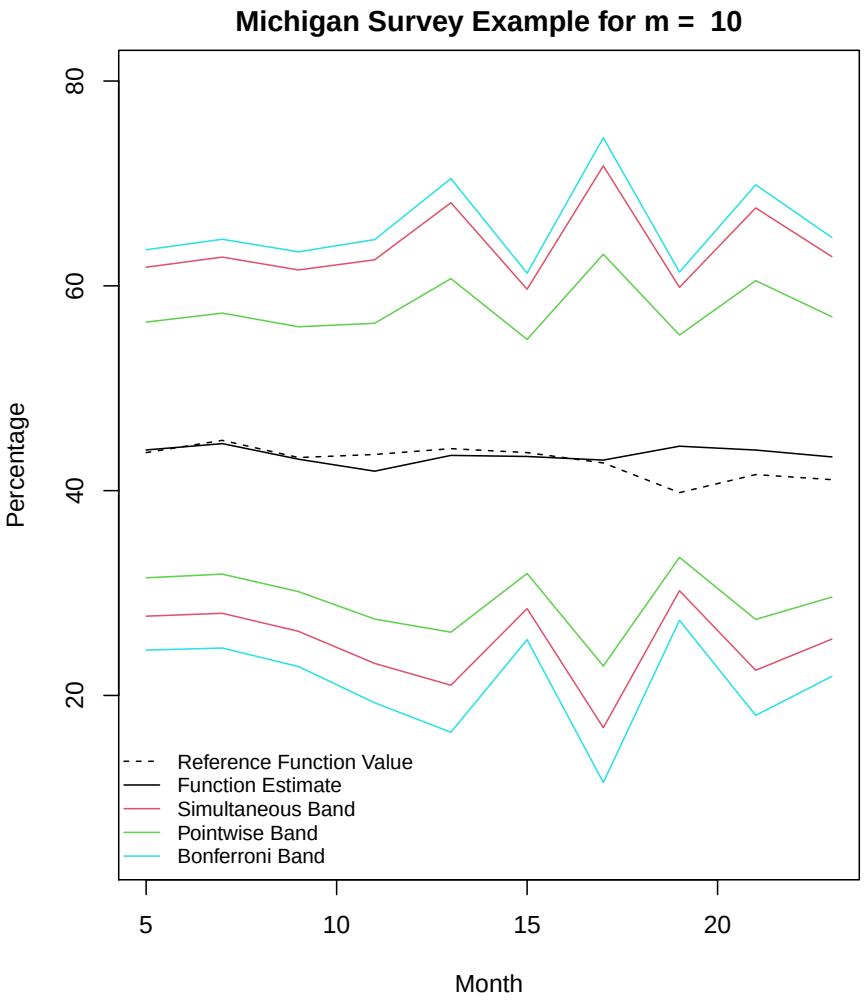


Figure 8.1: Michigan survey example: function estimates and confidence bands.

We want to estimate the variable value at the month number 5 to 23 for every 2 months with number of grid points $m = 10$. The reference value at these values are the average value from the dataset. The parameters we use are $n = 6879$ (the number of values from months left), $a_n = 117$ (number of batches), $b_n = 58$ (batch size). The bandwidth parameter $\lambda = 0.15$. We want to build simultaneous confidence band (component-wise method) for the targeted months at 80% level. The simulation was processed for just one time since there is no true function for the real data to evaluate against. We want to provide confidence bands for the months we are interested in.

The reference function value at a certain month is the average of all the responses at that month. As the reference function value is the average of responses of different people, the real data had a large variance. Our function estimate is close to the reference function value along the months. The simultaneous band is clearly separately from pointwise and Bonferroni bands. The confidence band widths are relatively wider for month 13, 17 and 21. The simultaneous band is averagely 36.5% wider than pointwise band. The Bonferroni band is averagely 14.8% wider than simultaneous band.

In this way, our methodology provides reasonable estimates and proper confidence regions for the percent that one thousand dollar investment will increase in value in the year ahead among the respondents. The bands we provided could be used for inferences.

8.2 Application: MAGIC Gamma Telescope Data Set

UCI machine learning repository has collected a variety of dataset with different types and lengths of data for research uses. One popular dataset is MAGIC Gamma Tele-

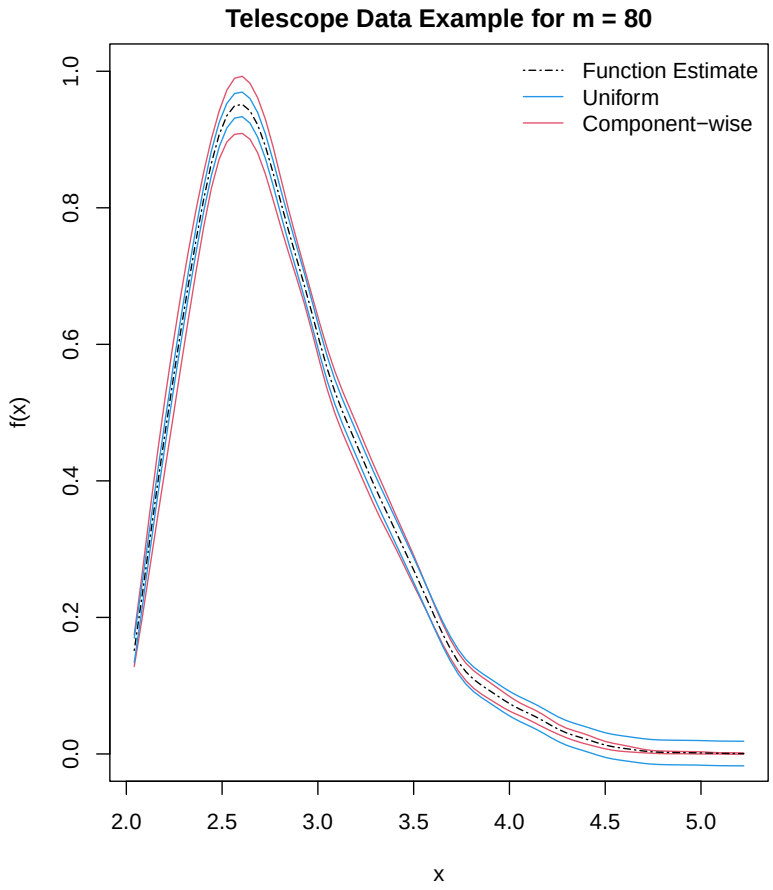


Figure 8.2: Telescope data example with uniform and simultaneous bands for $m = 80$.

scope Data Set (Dua and Graff, 2017). The data are Monte Carlo (MC) generated to simulate registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. The variable we explored is 'fSize', a continuous variable represents 10-log of sum of content of all pixels with length of $n = 19020$. The number of batches is $a_n = \lfloor \sqrt{\frac{n}{2.5}} \rfloor = 87$ and the batch size is $b_n = \lfloor \frac{n}{an} \rfloor = 218$. We estimated the density of 'fsize' on 80 points equally spaced from 2 to 5.5 with bandwidth $h = 0.075$. The simulation is done only once since we do not know the true density.

Table 8.1: Telescope Example relative width comparisons.

Method	Pointwise	Bonferroni	Component-wise	Optimization	Supt	Uniform
Width	1	1.962	1.724	1.8	1.755	5.655

Plot 8.2 shows the confidence bands generated from different methods with confidence level 90%. The uniform method provided the bands uniformly along all the points with the same width (Cheng and Chen, 2019). The component-wise method provides various bands at different points which is more reasonable as the measurement usually varies from point to point. The uniform and component-wise methods have similar bands except for the peak point (around $x = 2.7$) and the right boundary points ($x > 4$). The possible reason is that the variability at such points is relatively large or small than other points so the component-wise method is able to capture the variety change while the uniform method can not. From Table 8.1, the widths of uniform band are wider than other confidence bands. One huge advantage from our methodology is the time efficiency. Our methodology is 10 times faster than the uniform method which builds the uniform confidence band, and is described in the following section.

8.2.1 Uniform confidence band

Cheng and Chen described their way in building the uniform confidence bands of density function (Cheng and Chen, 2019). The algorithm is shown below. In step 4, the approximation is not quite computing efficient.

Algorithm 6 Uniform confidence band.

1: Select the bandwidth h from a cross-validation procedure.

2: Get the Kernel Density Estimator (KDE) $\hat{f}$.

3: Bootstrap the original sample for B times and compute the bootstrap KDE.

$$\hat{f}^{*(1)}, \hat{f}^{*(2)}, \ldots, \hat{f}^{*(B)}.$$

4: Compute the quantile

$$\hat{t}_{1-\alpha} = \hat{F}^{-1}(1-\alpha),$$

where

$$\hat{F}(t) = \frac{1}{B}\sum_{j=1}^{B} I(||\hat{f}^{*(j)} - \hat{f}||_\infty < t).$$

5: Obtain the uniform confidence band

$$\hat{C}_{1-\alpha}(x) = [\hat{f}(x) - \hat{t}_{1-\alpha}, \hat{f}(x) + \hat{t}_{1-\alpha}].$$

8.3 Time varying model example

Here is the general form for a linear time-varying coefficient model (Wu et al., 1998):

$$Y_{ij} = X_i^T(t_{ij})\beta(t_{ij}) + \epsilon_i(t_{ij}), \tag{8.1}$$

where $\epsilon_i(t)$ are stochastic processes with mean 0. $X_i(t) = (1, X_{i1}(t), \ldots, X_{ik}(t))^T$ and $\epsilon_i(t)$ are independent for $t_{ij} \in \mathbb{R}$. Set $\beta(t) = (1, \beta_0(t), \ldots, \beta_k(t))^T$, $\beta_l(t) \in \mathbb{R}$ for all $l = 0, \ldots, k$.

In the example, $X = (1, X_1, X_2, X_3)$ is a time-independent covariate. X_1 and X_2 are two Bernoulli random variables and X_3 is a $N(0, .25^2)$ random variable. The coefficient curves are given by

$$\beta_0(t) = 15 + 20 \sin\left(\frac{t\pi}{60}\right),$$

$$\beta_1(t) = 4 - \left(\frac{t-20}{10}\right)^2,$$

$$\beta_2(t) = 2 - 3\cos\left[\frac{(t-25)\pi}{15}\right]^2, \text{ and}$$

$$\beta_3(t) = -5 + \frac{(30-t)^3}{5000}.$$

There are 200 subjects selected randomly. The $X_i, i = 1, \ldots, 200$ represents a subject. Each subject was generated for X_1, X_2, X_3 from the following joint density

$$f(x_1, x_2, x_3) = \frac{0.5}{(2\pi)^{0.5}} \exp\left(-2x_3^2\right) 1_{[0,1]}(x_1) 1_{[0,1]}(x_2) 1_{[-\infty,\infty]}(x_3).$$

There were 30 equally spaced artificial "scheduled" time points and $n = 200$ random displacement points s_{i1} from the $U(0, 1)$ distribution so that $s_{il} = s_{i1} + (l-1)$, $l = 1, \ldots, 30$. For the purpose of randomness, each "scheduled" time point s_{il} was randomly missed with a probability of 60%. Then the remaining observed time points were denoted by t_{ij}.

In this way, each subject has unequal numbers of repated measurements n_i and different observed time points t_{ij}. The random errors $\epsilon_i(t_{ij})$ were simulated by a Guassian process with mean 0 and a covariance matrix

$$cov[\epsilon_{i_1}(t_{i_1j_1}), \epsilon_{i_2}(t_{i_2j_2})] = \begin{cases} 4\exp\left(-|t_{i_1j_1} - t_{i_2j_2}|\right) & \text{if } i_1 = i_2; \\ 0 & \text{if } i_1 \neq i_2. \end{cases}$$

Then the outcomes Y_{ij} were obtained by combining the observed $(t_{ij}, X_i, \epsilon_i(t_{ij}))$ and the coefficient functions into (8.1).

8.3.1 Wu's procedure

We want to estimate $\beta(t) = (\beta_0(t), \beta_1(t), \beta_2(t), \beta_3(t))$. For $i = 1, \ldots, n$,

$$X_i = \begin{pmatrix} 1 & X_{i1}(t_{i1}) & X_{i2}(t_{i1}) & X_{i3}(t_{i1}) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_{i1}(t_{in_i}) & X_{i2}(t_{in_i}) & X_{i3}(t_{in_i}) \end{pmatrix},$$

$$Y_i = \begin{pmatrix} Y_{i1} \\ \vdots \\ Y_{in_i} \end{pmatrix},$$

and $K_i(\cdot;\cdot)$ is a diagonal kernel matrix which is defined by

$$K_i(t;h) = diag\left(K\left(\frac{t - t_{i1}}{h}\right), \ldots, K\left(\frac{t - t_{in_i}}{h}\right)\right).$$

A least square estimator for $\beta(t)$ is

$$\hat{\beta}(t;h) = \left(\sum_{i=1}^{n} X_i^T K_i(t;h) X_i\right)^{-1} \left(\sum_{i=1}^{n} X_i^T K_i(t;h) Y_i\right). \tag{8.2}$$

By minimizing the average prediction squared error(APSE), the bandwidth h is given by cross-validation. From the asymptotic properties of $\hat{\beta}(t;h)$, the true function $\beta(t_0)$ could be

approximated by the local estimator $\hat{\beta}(t_0;h)$. With proper normalization, $\hat{\beta}(t_0;h)-\beta(t_0)$ has a multivariate Gaussian distribution suggested (Moyeed and Diggle, 1994). The asymptotic confidence regions could be built then. The pointwise confidence intervals could be given with or without bias correction. The confidence bands are Bonferroni-Type.

8.3.2 Building simultaneous bands

From (8.2), our method in estimating the variance-covariance matrix Σ was modified. We could regard the way in simulating $(t_{ij}, X_i, \epsilon_i(t_{ij}))$ as Markov Chains with Markov CLT by the Monte Carlo error, $\hat{\beta}_j(t;h) - \beta_j(t;h)$ for jth β element. Suppose we want to estimate β_j for $a = t_1 < t_2 < \cdots < t_m = b$. There is a $m \times m$ positive definite matrix Σ_j satisfying

$$\sqrt{n}\left(\hat{\beta}_j(t;h) - \beta_j(t;h)\right) \xrightarrow{\mathrm{d}} N_m(0, \Sigma_j), \tag{8.3}$$

when $n \to \infty$.

Similar as before, we denote $n = a_n b_n$, where a_n is the number of batches and b_n is the corresponding batch size. Unlike the partitioning procedure in the Markov Chains, we use the resampling procedure: a subset from n with size b_n was selected without replacement. Then the estimates of $\hat{\beta}_j(t;h)$ was calculated from this subset from (8.2), called $\tilde{\beta}_{j(k)}(t;h)$ with $k = 1, \ldots, a_n$. It means that the procedure was repeated with a_n times.

A batch mean estimator of Σ_j is given by

$$\Sigma_j = \frac{b_n}{a_n - 1} \sum_{k=0}^{a_n - 1} \left(\hat{\beta}_j(t;h) - \tilde{\beta}_{j(k)}(t;h)\right)\left(\hat{\beta}_j(t;h) - \tilde{\beta}_{j(k)}(t;h)\right)^T. \tag{8.4}$$

Table 8.2: Time varying model example: Running time comparisons for actions on bias correction.

Method	No bias correction	Mean function only	Bias correction
Running time	1 minute	2.5 hours	47 hours

Then the upper bound and lower bound could be achieved using our simultaneous band method described before.

8.3.3 Effectiveness of bias correction

The bias correction procedure is an important action in the method. So we will compare how the bias correction will affect the running time and the region width. There are three ways in applying the bias correction procedure: no bias correction, bias correction for the mean function $\hat{\beta}_j(t;h)$ only, bias correction for both the mean function and the batch mean estimators. From Table 8.2, the confidence bands among these methods are quite similar (within 0.001% differences). So the bias correction is not necessary in this situation. The estimators we will use are without the bias correction procedure.

8.3.4 Results

With number of points $n = 4,000$, the number of batched is $a_n = 40$ and the batch size $b_n = 100$, the simulation is repeated with 200 replications for 95% confidence level. The number of time points is $m = 28$ equally spaced from 1.5 to 28.5. We compare the pointwise method (Wu's default approach) and the simultaneous method (optimization). We do not plot the Bonferroni band which has a similar band width with the simultaneous band.

Table 8.3: Time-varying model: overall coverage comparisons among different confidence bands.

Function	β_0	β_1	β_2	β_3
Pointwise	0.135	0.335	0.2	0.290
Bonferroni	0.725	0.920	0.850	0.89
Optimization	0.7	0.91	0.84	0.885

Table 8.4: Time-varying model: relative widths comparisons among different confidence bands.

Function	β_0	β_1	β_2	β_3
Pointwise	1	1	1	1
Bonferroni	1.594	1.594	1.594	1.594
Optimization	1.515	1.525	1.528	1.533

From Table 8.3, the Bonferroni and optimization methods achieved the desired coverage 95% on β_1 while still needed more improvements in other $\beta_i's$. For β_1, the Optimization band has a MCSE 0.02 and a coverage confidence interval $[0.87, 0.95]$ which covers the desired probability. For β_3, the Optimization band has a MCSE 0.023 and a coverage confidence interval $[0.841, 0.929]$ which is a little far from 90%. The reason behind the relatively low coverage probability for β_0 and β_2 may result from inaccurate estimators or failing to capture the variability in the estimators.

From Table 8.4, the optimization bands are not quite narrow compared to the Bonferroni bands. For example, for β_1, the Bonferroni band is only 4.5% wider than the optimization band. This is maybe because the number of grid points is $m = 28$, not enough large to show the simultaneous property for the optimization method. Since we need to make sure the time is a whole number, we could not increase the number of grid points m

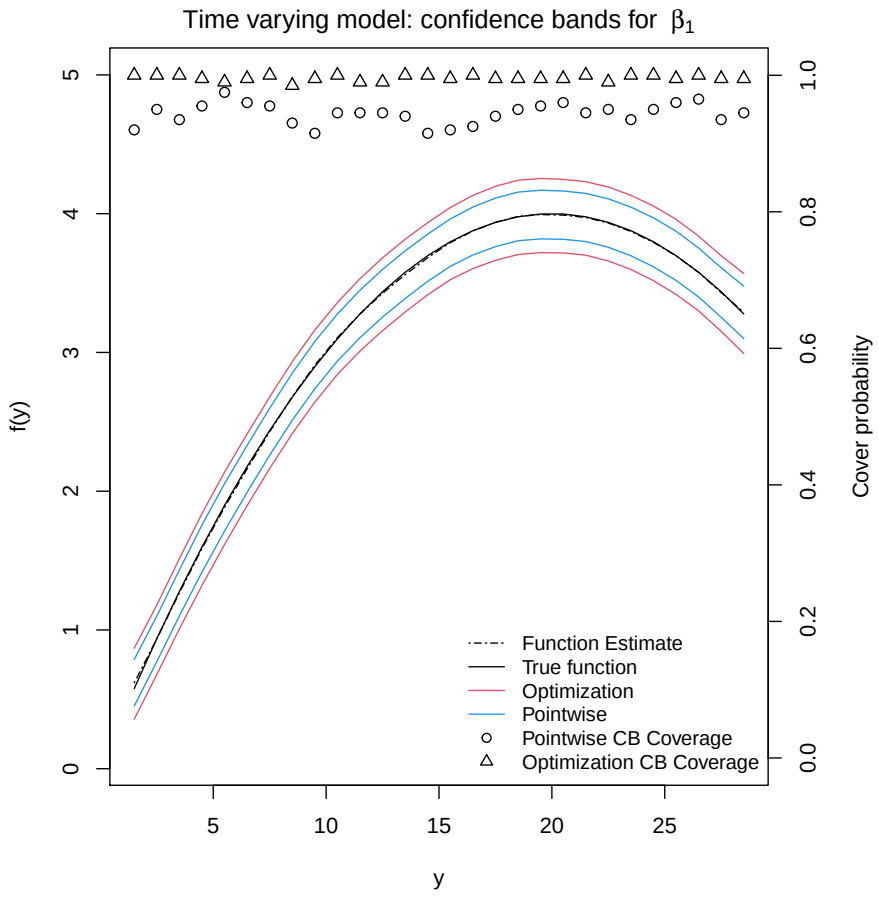


Figure 8.3: Time varying model confidence bands for β_1.

any more. From plots 8.3 and 8.4, we could tell that the coverage at each point is even smaller than 95% for pointwise band. We need to improve the estimator accuracy or provide a better covariance matrix in the future.

8.3.5 Discussion

Overall our simultaneous bands have reached the 95% desired coverage for β_1 as expected. Our simultaneous methodology still need some improvement for other $\beta_i's$. Prob-

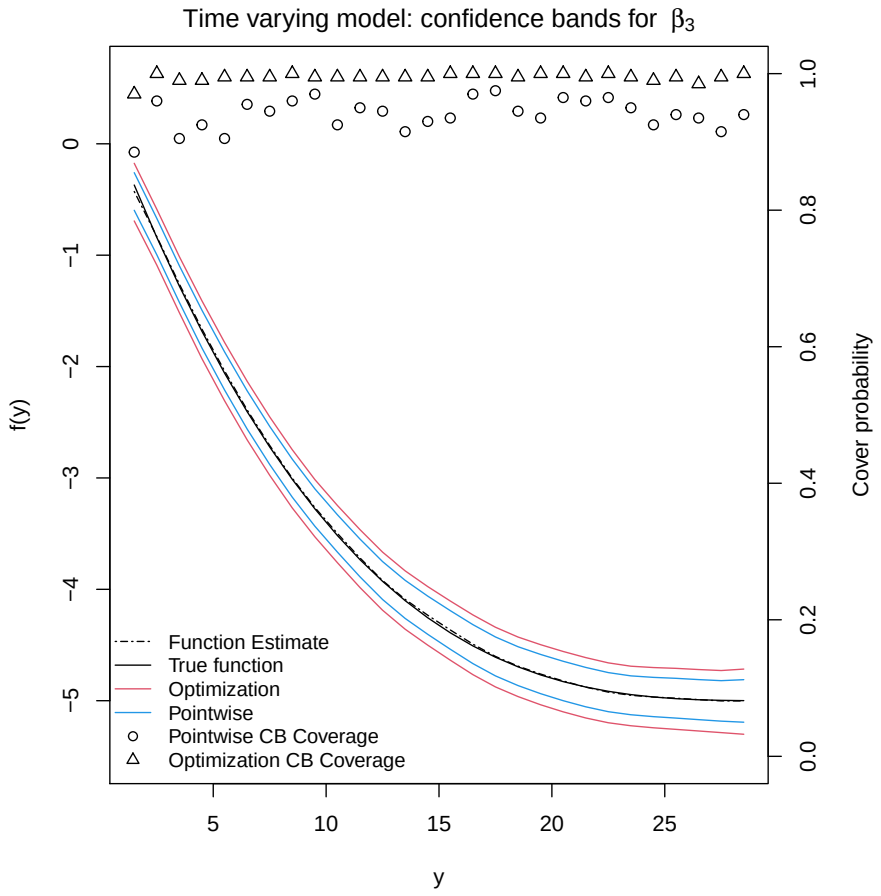


Figure 8.4: Time varying model confidence bands for β_3.

ably the reason is that there are still some covariances in the model which our simultaneous methods have not caught.

8.4 Application to a Bayesian reliability model

A Bayesian reliability model is discussed in Vats et al. (2020). For the purpose of testing the manufacturer's claim of expected lamp life in an Liquid crystal display (LCD) projector being 1,500 hr, identical lamps were placed in 31 projectors for various models and their time to failure was recorded. The table shows the data. For $i = 1, \ldots, 31$, let t_i denote the observed failure time for each lamp. We assumed that the t_i's are a realization from,

Table 8.5: Liquid crystal display time to failure in projection hours for 31 projectors.

387	182	244	600	627	332	418	300
798	584	660	39	274	174	50	34
1895	158	974	345	1755	1752	473	81
954	1407	230	464	380	131	1205	

$$T_i \sim Weibull(\lambda, \beta),$$

where $\lambda > 0$ is the scale parameter and $\beta > 0$ is the shape parameter for the Weibull distribution. Our interest is to estimate the reliability function. Under the Weibull likelihood, the reliability function is

$$R(t) = \exp\left(-\lambda t^{\beta}\right).$$

It is assumed that priors $\lambda \sim Gamma(2.5, 2350)$ and $\beta \sim Gamma(1, 1)$, where each is represented by a shape and rate parameter. The density of the posterior is

$$f(\lambda, \beta) \propto \lambda^{32.5} \beta^{31} \left(\prod_{i=1}^{31} t_i \right)^{\beta - 1} \exp\left\{ \lambda \sum_{i=1}^{31} t_i^{\beta} \right\} \exp\{-\beta\} \exp\{-2350\lambda\},$$

where the normalizing constatnt is unknown and we apply component-wise MCMC methods (Johnson et al., 2013) to sample from this distribution. The component λ can be updated by a Gibbs step and β will be updated through a Metropolis-Hasting step. The full conditional distribution of λ is

$$\lambda | \beta, T \sim Gamma(33.5, 2350 + \sum_{i=1}^{31} t_i^{\beta}).$$

Since the full condition distribution for β is not available in closed form, We implement a Metropolis-Hastings step from a Metropolis-within-Gibbs sampler (Robert and Casella, 2013). The proposal distribution is a $N(\cdot, 0.1^2)$, which yields an approximately optimal acceptance probability as suggested (Roberts et al., 1997). We update λ first, then β so only a starting value for β is needed. The starting value for β is the MLE, 1.12.

Assume we want to estimate and build confidence bands for the marginal posterior densities of λ and β on $m = 80$ points. For λ, the points are equally spaced between $(5 \times 10^{-4}, 3 \times 10^{-3})$ with selected bandwidth 3.04×10^{-5}. For β, the points are equally spaced between $(0.9, 1.3)$ with selected bandwidth 0.012. We generate two Markov chains for λ and β with $400,000$ draws each. For each point t_i, the batch size is $b_n = 400$ and the number of batches is $a_n = 1000$. The density estimator $\hat{f}_n(y)$ and the batch estimators $\{\tilde{f}_k(y), k = 1, \ldots, an\}$ were calculated using Rao-Blackwellized estimator and

kernel density estimator separately for the marginal posterior density of λ. Since the full condition distribution for β is not available, we only estimate the marginal posterior density of β using kernel density estimators. The confidence bands include pointwise, Bonferroni and simultaneous (Supt).

8.4.1 Results

Plot 8.5 showed different density estimators for the marginal posterior density of λ. As a parametric estimator, Rao-Blackwellized estimator has a little smoother estimation for the density compared to kernel density estimators. For kernel density estimators, the estimated curve are quite similar with or without bias correction.

From Plot 8.6, the confidence bands for the marginal posterior density of λ using Rao-Blackwellized estimators are given. The simultaneous band has a clear boundary from the pointwise band and Bonferroni band with both estimators. The confidence band is wider near the peak point (around $\lambda = 0.0008$). From Plot 8.7, the confidence bands for the marginal posterior density of λ using kernel density estimators (with bias correction) are given. Bonferroni band and simultaneous band are similar in band widths.

From Table 8.6, the confidence bands from Rao-Blackwellized estimators are narrower than these from kernel density estimators of same types. It is because the non-parametric estimator like kernel density estimators created more varieties in the covariance matrix to generate a wider band than the parametric estimator as the Rap-Blackwellized estimators did.

From Table 8.7, we could find the relative widths for different confidence bands for the marginal posterior density of λ. The relative width for the simultaneous band

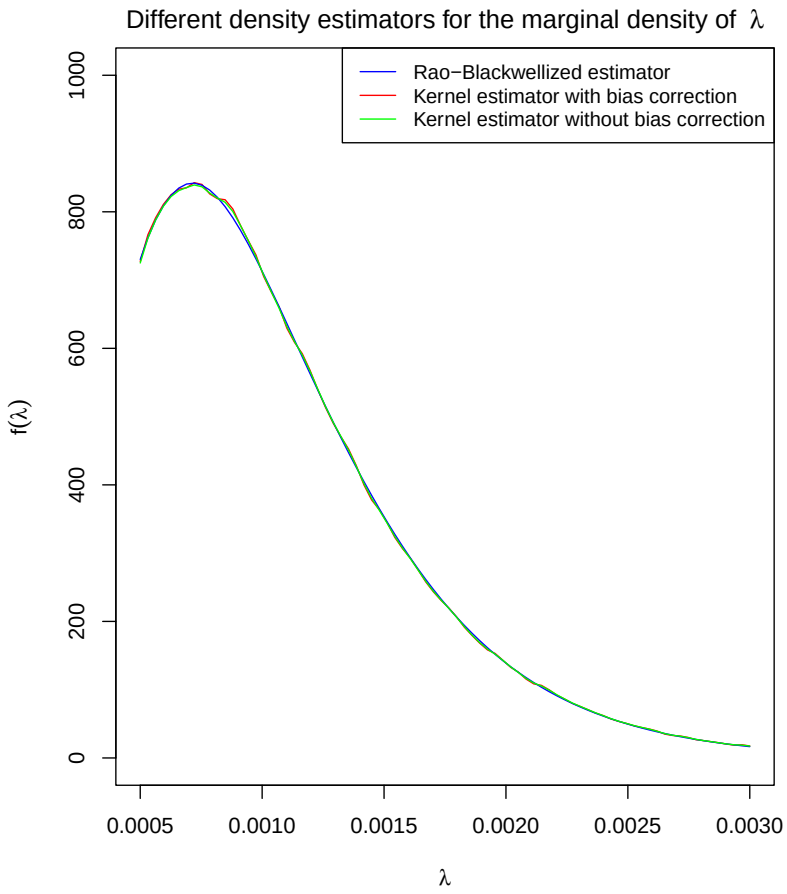


Figure 8.5: Bayesian reliability model: different density estimators for the marginal posterior density of λ.

is 1.909 using the kernel density estimator, which is larger than 1.489 which is obtained using the Rao-Blackwell estimator estimator. Bonferroni band is 2.8% wider than the simultaneous band using the kernel density estimator and 31.8% wider using the Rao-Blackwellized estimator.

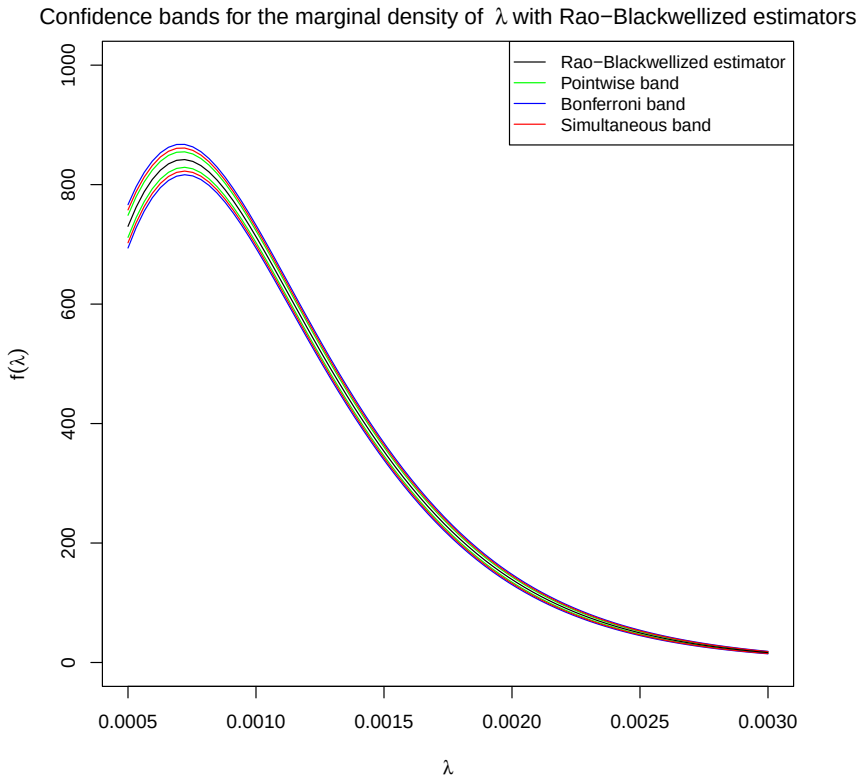


Figure 8.6: Bayesian reliability model: different confidence bands for the marginal posterior density of λ using Rao-Blackwellized estimators.

Table 8.6: Bayesian reliability model: band width comparisons among different methods for the marginal posterior density of λ.

Width	Pointwise	Bonferroni	Simultaneous
Rao-Blackwellized Estimator	13.0	25.4	19.3
Kernel Density Estimator	16.8	32.9	32.0

Table 8.7: Bayesian reliability model: relative band width comparisons among different methods for the marginal posterior density of λ.

Relative Width	Pointwise	Bonferroni	Simultaneous
Rao-Blackwellized Estimator	1	1.962	1.489
Kernel Density Estimator	1	1.962	1.909

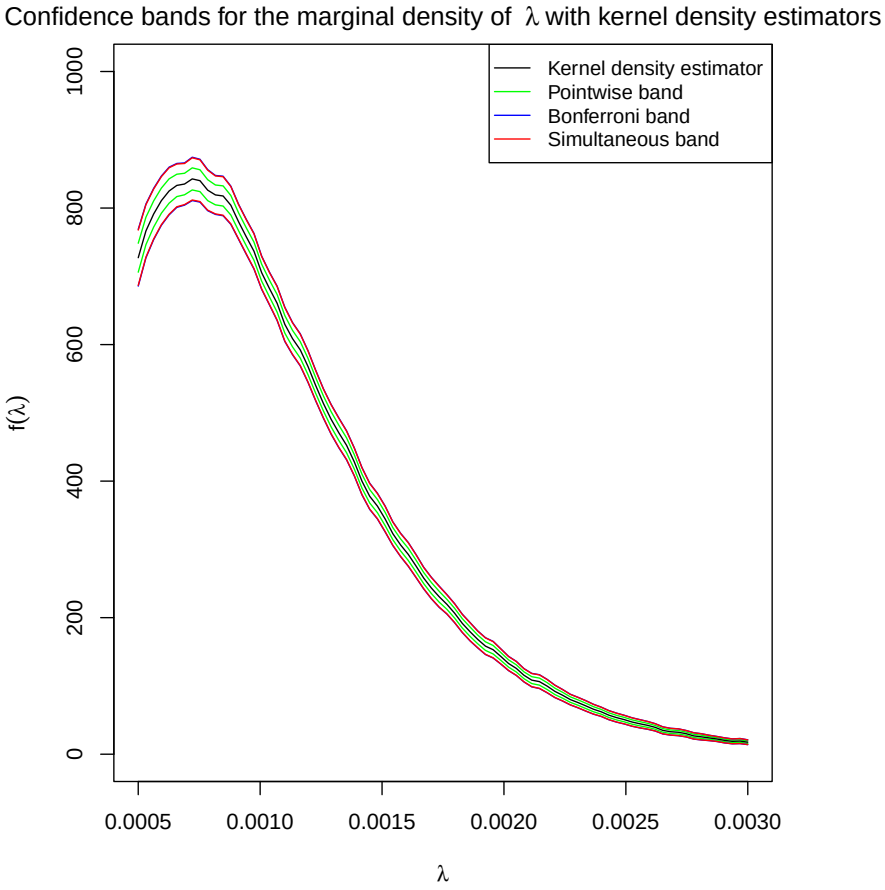


Figure 8.7: Bayesian reliability model: different confidence bands for the marginal posterior density of λ using kernel density estimators.

Table 8.8: Bayesian reliability model: band width comparisons among different methods for the marginal posterior density of β.

Method	Pointwise	Bonferroni	Simultaneous
Width	0.155	0.304	0.26
Relative Width	1	1.962	1.68

Plot 8.8 showed kernel density estimators for the marginal posterior density of β. The difference between the estimator with bias correction and estimator without bias correction lies near $\beta = 1.1$ where the second derivative is remarkable. From Plot 8.9, the confidence bands for the marginal posterior density of β using kernel density estimators (with bias correction) are given. The confidence bands are wider near $\beta = 1.1$.

From Table 8.8, the widths and relative widths for different confidence bands for the marginal posterior density of β are given. Bonferroni band is 16.8% wider than the simultaneous band.

Overall, we could provide various simultaneous confidence bands for the marginal posterior densities λ and β with different estimators for further inferences.

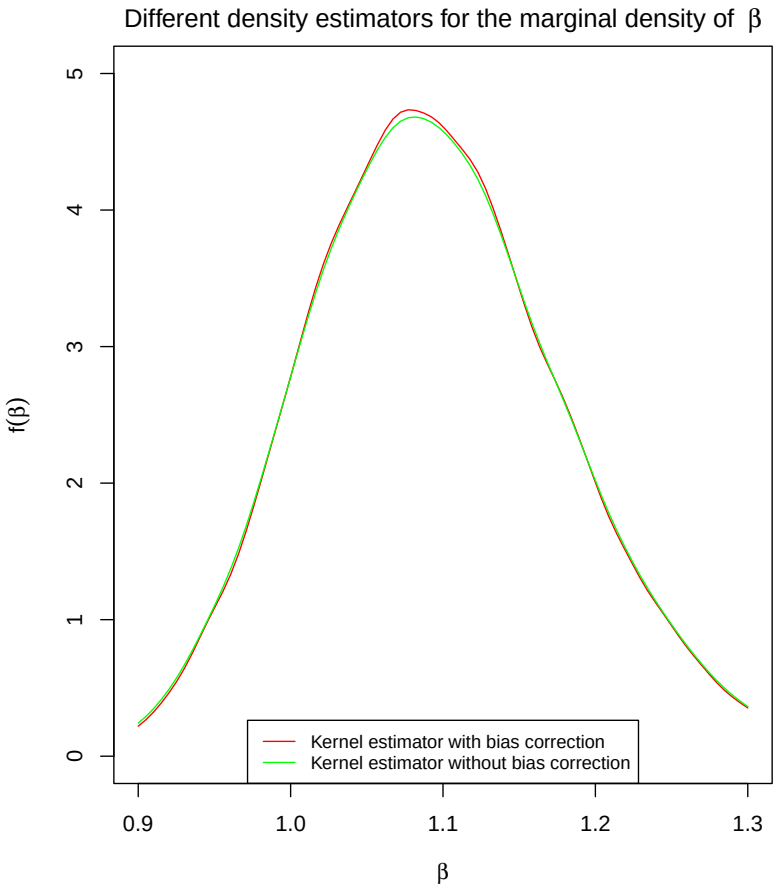


Figure 8.8: Bayesian reliability model: different density estimators for the marginal posterior density of β.

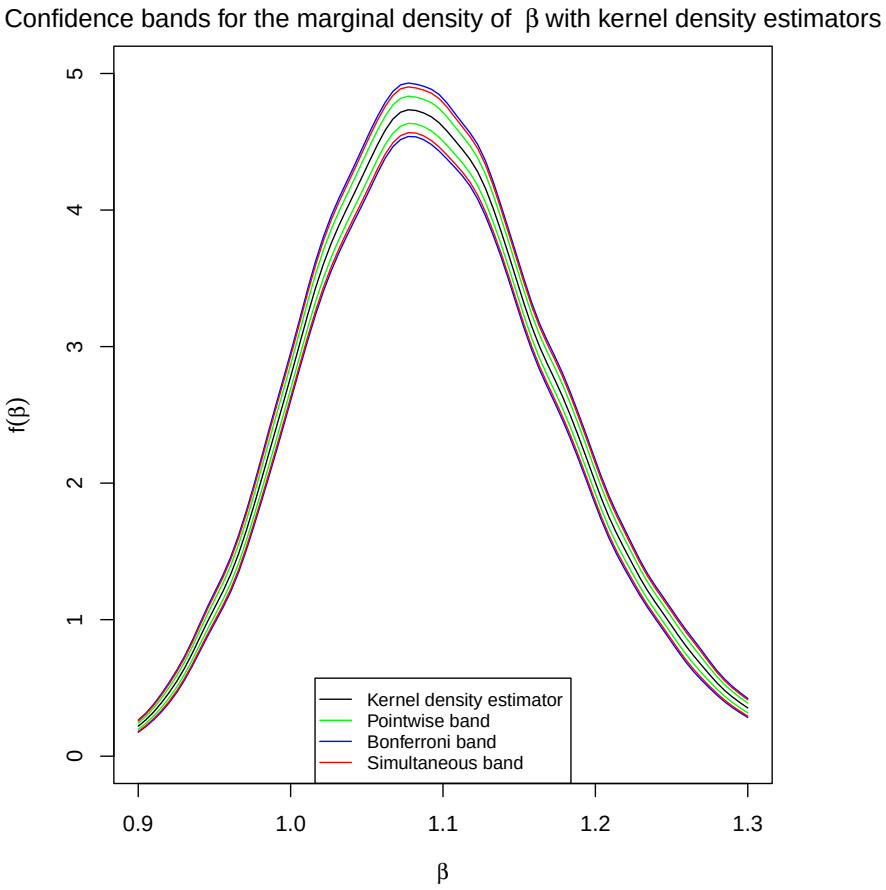


Figure 8.9: Bayesian reliability model: different confidence bands for the marginal posterior density of β using kernel density estimators.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

There are mainly three contributions of this dissertation. The first contribution is to quantify the uncertainty for functions from the MCMC settings. Though there have been various discussions in building confidence bands for functions, these work which usually have the independence assumptions lack the applications in the MCMC samples. Our methodology broadens the scope of confidence bands to the MCMC field.

Another contribution is to provide the bias correction procedures not only for the function estimators but also for the batch estimators. Usually the correction for bias is only conducted for the function estimator to improve the estimation accuracy. The bias correction approach for the batch estimators in our methodology captured the variability in the estimators and therefore provided more accurate variance-covariance estimators. The confidence bands would be more appropriate.

One great contribution is to summarize and evaluate different confidence bands in MCMC settings with simulations and real data examples. Pointwise, Bonferroni and various simultaneous confidence methods are performed and compared in different examples. These discussions provided some basics in evaluating different confidence band methods and inferred more investigations for quantifying the uncertainty with various approaches.

9.2 Future Work

9.2.1 Boundary points

Since our focus was on the simultaneous coverage for the functions, the effects of boundary points were not negligible. For example, in our simulation example

$$f(x) = 0.3e^x + 2\sin(x) - 2x,$$

where $x's$ are simulated from $Uniform(-1, 1)$. The $y's$ were generated by $y_i = f(x_i) + \epsilon_i$ with the noise $\epsilon_i \sim N(0, 0.5^2)$. The number of points to be estimated is $m = 20$ and the points are equally spaced between $(-0.8, 0.8)$. The bandwidth for the kernel $K_\lambda(x_0, x)$ is $\lambda = 0.15$. The confidence level is 90%. The simulation time is 200. For the pointwise band, the coverage for each point is shown below.

From Plot 9.1, the pointwise coverages for two boundary points $x = -0.8$ and $x = 0.8$ are 0.78 and 0.795. These coverages on the boundary points are much lower than other points whose average coverage is 0.943. Such effects also occurred in Bonferroni and simultaneous methods.

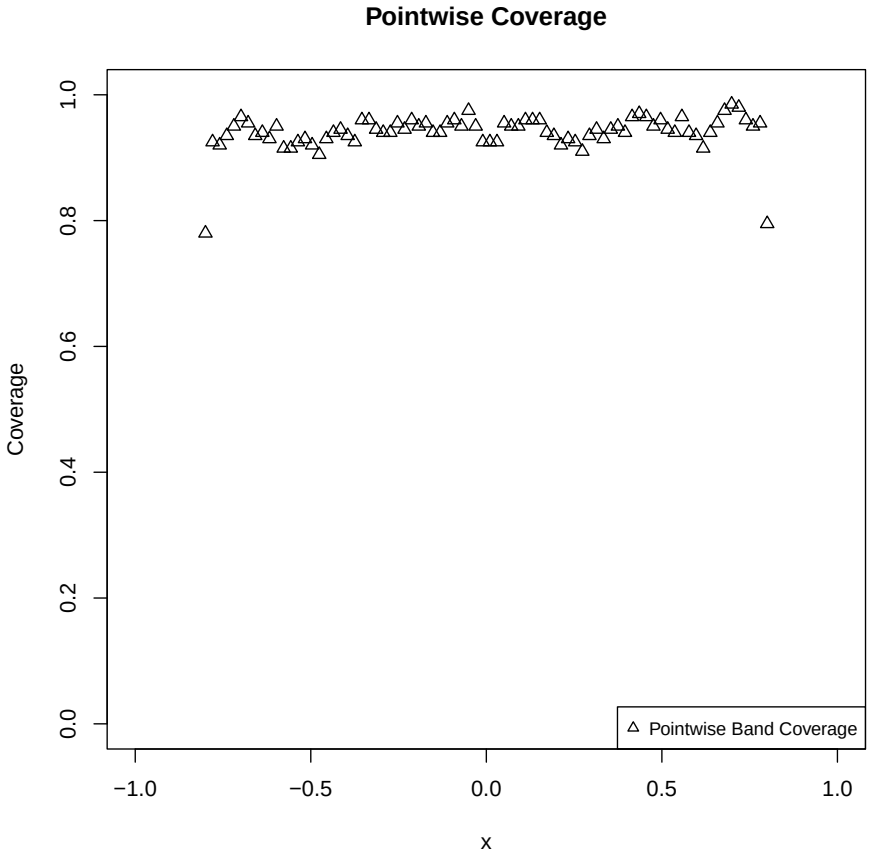


Figure 9.1: The effectiveness of boundary points on the pointwise coverage.

Table 9.1: Boundary points effects: total coverage probability.

Method	Pointwise	Bonferroni	Component-wise	Optimization	Supt
With boundary points	0.255	0.93	0.9	0.92	0.915
Without boundary points	0.4	0.98	0.96	0.98	0.97

From Table 9.1, the pointwise band coverage probability increased from 0.255 to 0.4 if the boundary points were removed. The Component-wise band coverage probability increased from 0.9 to 0.96 if the boundary points were removed. If we could improve the coverage on these boundary points, the overall coverage would increase and therefore provides better confidence bands.

9.2.2 Various estimation methods

In our methodology, we used the kernel density estimation and local linear regression method. Our estimation have shown some accuracy in estimating the true function and correcting the bias. There are still potential improvement in the estimation methods.

For density estimations, the estimation for the densities at the boundary points remained a problem (Zambom and Ronaldo, 2013). Local likelihood density estimation used a boundary kernel to solve this problem (Sheather, 2004).

For general function estimations, the locally weighted regression (loess) is a popular approach providing smooth fits for functions (Cleveland, 1979). The local linear regression we applied in our methodology is among the loess methods. There are also loess methods such as Local Polynomial Regression.

9.2.3 Lugsail batch mean estimator

Another potential improvement for building the confidence bands is about the batch estimator (Vats et al., 2019). The batch mean (BM) estimator in (5.2) does not use a lag window. For $r \geq 1$ and $0 \leq c \leq 1$, the lugsail lag window is

$$w_n(l) = \frac{1}{1-c}\Big(1 - \frac{|l|}{b_n}\Big) I\Big(0 \leq |l| \leq b_n\Big) - \frac{c}{1-c}\Big(1 - \frac{|l|}{b_n/r}\Big) I\Big(0 \leq |l| \leq \frac{b_n}{r}\Big).$$

Liu and Flegal (2018) proposed a family of weighted BM estimators that generalizes the BM estimator. For $l = 1, \cdots, b_n$, let $a_l = \lfloor (n/l) \rfloor$, then the weighted BM estimator is

$$\hat{\Sigma}_w = \sum_{l=1}^{b_n} \frac{1}{a_l - 1} \sum_{k=0}^{a_k - 1} l^2 \Delta_2 w_n(l) \left(\tilde{f}_k(\overrightarrow{w}) - \hat{f}_n(\overrightarrow{w})\right) \left(\tilde{f}_k(\overrightarrow{w}) - \hat{f}_n(\overrightarrow{w})\right)^T, \tag{9.1}$$

where $\Delta_2 w_n(l) = w_n(l-1) - 2w_n(l) + w_n(l+1)$. By using $c = 0$ and $r = 1$, we get the BM estimator $\hat{\Sigma}_{bn}$ in (5.2) from (9.1). We get a lugsail BM estimator, $\hat{\Sigma}_L$ using the lugsail window. By calculation, the relationship between $\hat{\Sigma}_L$ and $\hat{\Sigma}_{bn}$ is

$$\hat{\Sigma}_L = \frac{1}{1-c}\hat{\Sigma}_{bn} - \frac{c}{1-c}\hat{\Sigma}_{bn/r}.$$

The suggested r and c values in lugsail method are $r = 2$ and $c = 0.5$.

For the following two-component mixed normal distribution function

$$0.5N(-1, (2/3)^2) + 0.5N(1, (2/3)^2), \tag{9.2}$$

where $N(\mu, \sigma)$ is a normal distribution function with mean μ and standard deviation σ. In order to get correlated draws, the data were sampled from Metropolis–Hastings algorithm.

The goal is to build the confidence bands for $m = 20$ grid point equally spaced along $(-2, 2)$ from different methods with confidence level 90%. In each simulation, $n = 400,000$ draws were generated. With the number of batches $an = 400$ and the batch size $bn = 1000$, the density estimator $\hat{f}_n(\overrightarrow{w})$ and the batch estimators $\{\tilde{f}_k(\overrightarrow{w}), k = 1, \ldots, an\}$ were calculated. Here is the result table for 200 simulations.

Table 9.2: Lugsail batch mean estimator: coverage probability comparisons for two-component mixed normal using different batch estimators

m	20	20	40	40
Method	BM	Lugsail	BM	Lugsail
Pointwise	0.105	0.105	0.015	0.015
Bonferroni	0.885	0.885	0.895	0.89
Component-wise	0.87	0.865	0.875	0.875
Optimization	0.88	0.87	0.88	0.88
Supt	0.87	0.88	0.88	0.88

The lugsail performs similarly to the original BM estimator. We still need more evidence to explore how the lugsail could behave when it comes to other functions.

9.2.4 Building simultaneous confidence bands for estimators of the mean functions

When it comes to building simultaneous confidence bands for the estimators for the mean of functions, there are some existing methodology from past researchers. Degras(2017) suggested a way in evaluating the uncertainty for estimators in functional analysis.

For a random process X defined on a continuous domain D, the mean function function μ is well-defined. In functional analysis, it is important to assess the behavior of

the function over the entire domain D. So the functional parameter θ is of great interest to explore by building simultaneous confidence bands with level $1-\alpha$.

The goal is to find such bands $\{[L_s(t), U_s(t)] : t \in D\}$ that covers the mean function μ with probability $1-\alpha$

$$P(\mu(t) \in [L_s(t), U_s(t)], \forall t \in D) = 1-\alpha.$$

Such bands are known as simultaneous confidence bands.

With a given stochastic process X, the decomposition contains the mean function $\mu = E(x)$ and a zero mean process $Z = X - \mu$

$$X(t) = \mu(t) + Z(t), t \in D,$$

From the data collection procedure, the noisy observations $X_1, \ldots, X_n$ of X could be obtained. Regardless of the data generating process or other factors, a general model could be considered.

$$Y_{ij} = X_i(t_{ij}) + \epsilon_{ij}, i = 1, \ldots, n, j = 1, \ldots, p_i,$$

where Y_{ij} represents the observation of the ith subject or statistical unit at location t_{ij} and ϵ_{ij} is a measurement error. The X_i are assumed to be mutually independent and have independence from ϵ_{ij}. The ϵ_{ij} are typically independent across units(i) while the inner correlation among ϵ_{ij} is not necessary.

It is certain to use the sample mean estimator $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ in an ideal situation where there are infinite size of observations and no measurement error. Unfortunately, it is impossible to achieve such perfect estimator in the real world.

The technique to deal with finite size of observations and the unavoidable measure error is nonparametric smoothing. The data are locally averaged with certain weights. The mean function μ could be estimated by the nonparametric smoothing

$$\hat{\mu}(t) = \sum_{j=1}^{p} w_j(t)\bar{Y}_j,$$

where w_j is the weight function and $\bar{Y}_j = \frac{1}{n}\sum_{i=1}^{n} Y_{ij}$ at grid point t_j. A classic nonparametric estimator is the Nadaraya–Watson kernel estimator defined before. The non-parametric smoothing techniques when $D \subset R$ include smoothing spline, basis functions, p-spline and local polynomial. The choice could depend on the special situation of the problem in the study.

It is critical to estiamte the covariance Γ of the process X. With all $s, t \in D$ and nonparametric estimators $\hat{\mu}$, the following approximation holds

$$Cov(\hat{\mu}(s), \hat{\mu}(t)) \approx \frac{\Gamma(s,t)}{n},$$

when $n, p \to \infty$. The presmoothed data data are

$$\hat{X}_i(t) = \sum_{j=1}^{p} w_j(t)Y_{ij},$$

where $w_j(t)$ is the weight function defined above. With the sample covariance of $\hat{X}_i(t)$, the estimation of Γ is

$$\hat{\Gamma}(s,t) = \frac{1}{n-1}\sum_{i=1}^{n}(\hat{X}_i(s) - \hat{\mu}(s))(\hat{X}_i(t) - \hat{\mu}(t)).$$

A functional central limit theorem states that

$$\sqrt{n}(\hat{\mu} - \mu) \xrightarrow{\mathrm{d}} \mathcal{G}(0, \Gamma),$$

as $n, p \to \infty$. The $\mathcal{G}(0, \Gamma)$ denotes a Guassian process with mean zero and covariance Γ. With all these settings, it is not difficult to build simultaneous confidence bands for μ. The $\sigma(t) = \Gamma(t,t)^{\frac{1}{2}}$ and $\rho(s,t) = \frac{\Gamma(s,t)}{\sigma(s)\sigma(t)}$ are the standard deviation and correlation functions at t.

For a certain confidence level $1-\alpha$, the quantile $z_{\alpha,p}$ could be determined by $P(sup_{t\in D}|Z(t)| \leq z_{\alpha,p}) = 1-\alpha$. The $1-\alpha$ level simultaneous confidence band for μ is

$$\{[\hat{\mu}(t) - z_{\alpha,p}\frac{\sigma(t)}{\sqrt{n}}, \hat{\mu}(t) + z_{\alpha,p}\frac{\sigma(t)}{\sqrt{n}}] : t \in D\}.$$

With large n and p, the following equation holds:

$$P\Big(\mu(t) \in [\hat{\mu}(t) \pm z_{\alpha,p}\frac{\sigma(t)}{\sqrt{n}}], \forall t \in D\Big) \approx 1-\alpha.$$

The way to determine the quantile $z_{\alpha,p}$ is regarded as a parametric bootstrap of the standardized estimator $\frac{\sqrt{n}}{\hat{\sigma}(t)}(\hat{\mu}(t) - \mu(t))$.

Algorithm 7. Calculate $\hat{z}_{\alpha,p}$ for building simultaneous confidence bands for the mean of functional data

1: Generate Z_m as the random vector from discretizing the process $Z \sim \mathcal{G}(0, \Gamma)$ over a fine grid $\tau = \{\tau_1, \ldots, \tau_m\} \subset D$.

2: Simulate $Z_m \sim N(0, M_{\hat{\rho}})$ where $M_{\hat{\rho}} = (\hat{\rho}(\tau_j, \tau_k))_{1 \leq j,k \leq m}$.

3: Get the l_∞ norm, $||Z_m||_\infty$ which is the maximum of the absolute values of its entries for N simulations.

4: $\hat{z}_{\alpha,p}$ is the quantile of level $1 - \alpha$ of the N simulated $||Z_m||_\infty$.

The whole methodology above provided the simultaneous confidence bands for the mean function. While our main focus is on the functions from MCMC simulations which is quite different, we are looking forward to applying our described approaches in quantifying the uncertainty of confidence bands of the mean function in the future.

Bibliography

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An Introduction to MCMC for Machine Learning. *Machine learning*, 50(1-2):5–43.

Calonico, S., Cattaneo, M. D., and Farrell, M. H. (2018). On the effect of bias estimation on coverage accuracy in nonparametric inference. *Journal of the American Statistical Association*, 113(522):767–779.

Casella, G. and Robert, C. P. (1996). Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94.

Chen, D.-F. R. and Seila, A. F. (1987). Multivariate inference in stationary simulation using batch means. In *Proceedings of the 19th conference on Winter simulation*, pages 302–304.

Chen, Y.-C. (2017). A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1(1):161–187.

Cheng, G. and Chen, Y.-C. (2019). Nonparametric inference via bootstrapping the debiased estimator. *Electronic Journal of Statistics*, 13(1):2194–2256.

Cleveland, W., Grosse, E., and Shyu, W. (1991). Local regression models. chapter 8 of statistical models in s (edited by jm chambers and tj hastie), 309-376.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836.

Curtin, R. T. (1982). Indicators of consumer behavior: The university of michigan surveys of consumers. *Public Opinion Quarterly*, 46(3):340–352.

Degras, D. (2017). Simultaneous confidence bands for the mean of functional data. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(3):e1397.

Deng, H. and Wickham, H. (2011). Density Estimation in R. *Electronic publication*.

Dominitz, J. and Manski, C. F. (2011). Measuring and interpreting expectations of equity returns. *Journal of Applied Econometrics*, 26(3):352–370.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Dunn, O. J. (1958). Estimation of the means of dependent variables. *The Annals of Mathematical Statistics*, pages 1095–1111.

Epanechnikov, V. A. (1969). Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158.

Fan, J. (1992). Design-adaptive nonparametric regression. *Journal of the American statistical Association*, 87(420):998–1004.

Flegal, J. M. and Jones, G. L. (2010a). Batch means and spectral variance estimators in markov chain monte carlo. *The Annals of Statistics*, 38(2):1034–1070.

Flegal, J. M. and Jones, G. L. (2010b). Chapter 1 Implementing Markov Chain Monte Carlo: Estimating with Confidence.

Hamada, M. S., Wilson, A., Reese, C. S., and Martz, H. (2008). *Bayesian reliability*. Springer Science & Business Media.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Izenman, A. J. (1991). Review Papers: Recent Developments in Nonparametric Density Estimation. *Journal of the American Statistical Association*, 86(413):205–224.

Johnson, A. A., Jones, G. L., Neath, R. C., et al. (2013). Component-wise markov chain monte carlo: Uniform and geometric ergodicity under mixing and composition. *Statistical Science*, 28(3):360–375.

Jones, G. L. (2004). On the Markov Chain Central Limit Theorem. *Probability surveys*, 1(299-320):5–1.

Jones, G. L., Haran, M., Caffo, B. S., and Neath, R. (2006). Fixed-width Output Analysis for Markov Chain Monte Carlo. *Journal of the American Statistical Association*, 101(476):1537–1547.

Jones, M., Linton, O., and Nielsen, J. (1995). A simple bias reduction method for density estimation. *Biometrika*, 82(2):327–338.

Kim, H. J., MacEachern, S. N., and Jung, Y. (2016). Bandwidth selection for kernel density estimation with a markov chain monte carlo sample. *arXiv preprint arXiv:1607.08274*.

Lantz, B. (2013). *Machine learning with R*. Packt publishing ltd.

Liu, Y. and Flegal, J. M. (2018). Weighted batch means estimators in markov chain monte carlo. *Electronic Journal of Statistics*, 12(2):3397–3442.

Matthews, J. H. (2008). Simpson's 3/8 rule for numerical integration. *Numerical Analysis-Numerical Methods Project. California State University, Fullerton. Archived from the original on*, 4.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Montiel Olea, J. L. and Plagborg-Møller, M. (2019). Simultaneous confidence bands: Theory, implementation, and an application to svars. *Journal of Applied Econometrics*, 34(1):1–17.

Moyeed, R. and Diggle, P. J. (1994). Rates of convergence in semi-parametric modelling of longitudinal data. *Australian Journal of Statistics*, 36(1):75–93.

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

Politis, D. N., Romano, J. P., and Wolf, M. (1999). *Subsampling*. Springer Science & Business Media.

Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

Roberts, G. O., Gelman, A., Gilks, W. R., et al. (1997). Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349–367.

Robertson, N., Flegal, J. M., Vats, D., and Jones, G. L. (2020). Assessing and visualizing simultaneous simulation error. *Journal of Computational and Graphical Statistics*, pages 1–11.

Sheather, S. J. (2004). Density estimation. *Statistical science*, pages 588–597.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(3):683–690.

Sun, J. and Loader, C. R. (1994). Simultaneous confidence bands for linear regression and smoothing. *The Annals of Statistics*, 22(3):1328–1345.

Vats, D. and Flegal, J. M. (2018). Lugsail lag windows and their application to mcmc. *arXiv preprint arXiv:1809.04541*.

Vats, D., Flegal, J. M., and Jones, G. L. (2019). Multivariate output analysis for Markov chain Monte Carlo. *Biometrika*, 106(2):321–337.

Vats, D., Robertson, N., Flegal, J. M., and Jones, G. L. (2020). Analyzing Markov chain Monte Carlo output. *Wiley Interdisciplinary Reviews: Computational Statistics*, 12(4):e1501.

Wasserman, L. (2006). *All of nonparametric statistics*. Springer Science & Business Media.

Wei, G. C. and Tanner, M. A. (1990). A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American statistical Association*, 85(411):699–704.

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(444):1388–1402.

Zambom, A. Z. and Ronaldo, D. (2013). A review of kernel density estimation with applications to econometrics. *International Econometric Review*, 5(1):20–42.