Bayesian Integration for Assessing the Quality of the Laplace Approximation

by

Charlie (Haoxuan) Zhou

B.Sc., Simon Fraser University, 2015

Project Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science

in the Department of Statistics and Actuarial Science

Faculty of Science

© Charlie (Haoxuan) Zhou 2017
SIMON FRASER UNIVERSITY
Spring 2017

All rights reserved. However, in accordance with the Copyright Act of Canada, this work may be reproduced without authorization under the conditions for “Fair Dealing.” Therefore, limited reproduction of this work for the purposes of private study, research, education, satire, parody, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.
# Approval

Name: Charlie (Haoxuan) Zhou  
Degree: Master of Science (Statistics)  
Title: *Bayesian Integration for Assessing the Quality of the Laplace Approximation*  
Examining Committee: Chair: Jinko Graham  
   Professor  
   David Alexander Campbell  
   Senior Supervisor  
   Associate Professor  
   Derek Bingham  
   Supervisor  
   Professor  
   Liangliang Wang  
   Examiner  
   Assistant Professor  

Date Defended: 24 November 2017
Abstract

Nuisance parameters increase in number with additional data collected. In dynamic models, this typically results in more parameters than observations making direct estimation intractable. The Laplace Approximation is the standard tool for approximating the high dimensional integral required to marginalize over the nuisance parameters. However the Laplace Approximation relies on asymptotic arguments that are unobtainable for nuisance parameters. The way to assess the quality of the Laplace Approximation relies on much slower MCMC based methods. In this work, a probabilistic integration approach is used to develop a diagnostic for the quality of the Laplace Approximation.

Keywords: Bayesian methodology, Partial Differential Equation, Gaussian Process in Machine learning
Dedication

To my parents Mr. Jiyang Zhou and Mrs. LiPing Wu, my family, and my friends.
Acknowledgements

I want to thank all the people who help me and accompany me during my Master program. First, I wish to express my deepest gratitude to my supervisor, Dr. David Alexander Campbell, who helps me to find this interesting research problem and guide me through the whole time in study and in life. Thank you for showing me how to build a methodology from scratch and how to use it in the real cases. I admire your passion on statistics and your knowledge. You teach me how to become a successful statistician, encourage me on my study, and help me on my writing. I cannot make this far without your help.

I also want to thank all the professors in the department of statistics and actuarial science. You are all be really kind to me. I want to thank my co-supervisor, Dr. Rick Routledge, who support me on funding and teach me how to look the problem at a bigger picture. I want to thank Dr. Oksana Chkrebtii, who help me understand her paper and share her MATLAB code with me. I am grateful to Dr. Tim Swartz, who opened me mind and showed me what is a Bayesian world looks like. I also want to thank Dr. John Hu, who help me build my confidence and encourage me going further on statistics. I gratefully acknowledge the support from Dr Jack Davis who helps me on proof reading. Finally, I would like to thank Dr. Derek Bingham and Dr. Liangliang Wang for taking time from their busy schedule to serve as my internal examiner.

I want to thank my fellow graduate students. Thank you Perry(Peijun) Sang and Yuping Yang for helping me understand the mathematical theories of many papers. Thank you JinChoeil Choi, Shufei Ge, Yan Lin, Dongmeng Liu, Moyan Mei, Pulindu Ratnasekera, Chenlu Shi, Biljana Jonoska Stojkova, Terry Tang, Trevor Thomson, Shijia Wang, Jiying Wen, Yi Xiong, and Joanna Zhao for showing up in my life and accompany me during my hard time. We have lunch together, we play board games together, we say ‘See you today/within 8 hours’ to each other everyday, and we spend more than half day in one room for the same questions. It was always fun to discuss life or statistical questions with you. I also want to thank Dongdong Li, who is a good friend since I was an undergraduate student in here. You are not only help me on study but also let me aware my shortcomings.

Finally, I want to thank my parents. Although they are not here, they always support my back. They are always on my side no matter I have good time or bad time.
# Table of Contents

Approval ........................................ ii
Abstract ........................................... iii
Dedication ........................................ iv
Acknowledgements .................................v
Table of Contents ................................ vi
List of Tables ....................................... viii
List of Figures ...................................... ix

1 Introduction ................................... 1
   1.1 Laplace Approximation ................................. 2
   1.2 Motivation ........................................... 2

2 Methodology .................................... 4
   2.1 Probabilistic Solver ................................. 4
   2.2 Probabilistic Solver Prior ............................ 4
   2.3 Updating the Mean and Covariance ................... 6
   2.4 The Covariance Kernels .............................. 6
   2.5 Probabilistic Solver in higher Dimensions .......... 7
   2.6 Tuning Parameters ................................... 8
   2.7 Simple Experiments .................................. 11
      2.7.1 Choice of Evaluation Grid ..................... 13

3 Applications ................................... 17
   3.1 One Dimensional Normal Distribution ............... 17
   3.2 Bivariate Normal Distribution ...................... 17
   3.3 T Distribution ..................................... 20
       3.3.1 One Dimensional T Distribution ............... 20
       3.3.2 Two Dimensional T distribution ................ 21
List of Tables

Table 2.1 The tuning result of $\alpha$ and $\lambda_c (\hat{\alpha} \text{ and } \hat{\lambda}_c)$ in one dimensional Cases . . 10
List of Figures

Figure 2.1 In panel 2.1a and panel 2.1b are realizations of the uniform kernel and the square exponential kernel when $t$ is from -1 to 1 respectively.

Figure 2.2 When the target distribution is an one dimensional standard normal distribution, $\alpha \in [1:100], \lambda_c \in [1:5]$, the behaviour of $D_{tuning}$.

Figure 2.3 When the target distribution is an one dimensional standard normal distribution, $\alpha \in [1:100], \lambda_c \in [1:5]$, the behaviour of $D_{tuningabs}$.

Figure 2.4 The first row represents the evaluated results and true values of the full function, target function, and the histogram of evaluated target function on the last evaluation point when the uniform kernel is used. The second row represents the corresponding evaluated results with the true values when the squared exponential kernel is used, and the vertical line is the LA value.

Figure 2.5 The first row represents the evaluated results with the true values of the full function, the target function, and the histogram of evaluated target function on the last evaluation point when the “fill in” method is used. The second row represents the corresponding evaluated results with the true values when the “standard deviation extend” method is used, and the vertical line is the LA value.

Figure 2.6 The histogram on the last evaluational point when “standard deviation extend” method for $c = 5, 7, 9, 11, 15$, and 17 respectively, the vertical line is the LA value.

Figure 2.7 The histogram on the last evaluational point when “fill in” method for $c = 5, 7, 9, 11, 15$, and 17 respectively, and the vertical line is the LA value.

Figure 3.1 The first row represents the evaluated results and true values of the full function, the target function, and the histogram of evaluated target function on the last evaluation point when the uniform kernel is used. The second row represents the corresponding evaluated results with true values when the squared exponential kernel is used, and the vertical line is the LA value.

Figure 3.2
Figure 3.3 ................................................................. 19
Figure 3.4 ................................................................. 20
Figure 3.5 The first row represents the evaluated results and the true values of
the full function, the target function, and the histogram of evaluated
target function on the last evaluation point when the uniform kernel
is used. The second row represents the corresponding evaluated
results with the true values when the squared exponential kernel is
used, and the vertical line is the LA value. ....................... 21
Figure 3.6 ................................................................. 22
Figure 3.7 ................................................................. 22
Figure 3.8 ................................................................. 23
Figure 3.9 The first row represents the evaluated results and true values of the
full function, the target function, and the histogram of evaluated tar-
target function on the last evaluation point when the uniform kernel is
used. The second row represents the corresponding evaluated results
with true values when the squared exponential kernel is used, and
the vertical line is the LA value. ....................... 24
Figure 3.10 ............................................................... 25
Figure 3.11 ............................................................... 25
Figure 3.12 ............................................................... 26
Figure 3.13 The first row represents the evaluated target function of normal, T,
and gamma distribution with the uniform kernel respectively, and
the second row represents the evaluated target functions of normal,
T, and gamma distribution with the square exponential kernel re-
spectively, and the vertical line is the LA value. .............. 27
Figure 3.14 The first row represents the evaluated the target functions of nor-
mal, T, and gamma distribution with uniform kernel respectively,
and the second row represents the evaluated the target functions of
normal, T, and gamma distribution with square exponential kernel
respectively, and the vertical line is the LA value. ........... 28
Chapter 1

Introduction

Nuisance parameters increase in number with additional data collected. In dynamic models, this results in more parameters than observations making direct estimation intractable. Typical strategies involve treating nuisance parameters as random effects in an EM formulation where evaluation of the expectation requires a high dimensional integral over the nuisance parameters or directly marginalizing out the nuisance parameters while optimizing over the structural parameters. When the required integral is analytically intractable, numerical methods can be applied over the high dimensional space. The computational power required to perform integration of a sufficiently dense grid of high dimensional points makes numerical solvers infeasibly slow. Using a less dense grid makes the numerical integration computationally feasible but results in a biased integral. Although, Monte Carlo integration is an option, the high dimensional nature of the problem makes it infeasibly slow to test and iterate a variety of model formulations.

Laplace approximations use a second order Taylor approximation to the nuisance parameter portion of the likelihood, which, under asymptotic arguments result in an analytic expression for the integral. The speed of the approach has led to the wide spread success of the Laplace Approximation and its versatility has seen its expansion into Bayesian methods via Integrated Nested Laplace Approximation [2]. Despite its success, the Laplace approximation relies on asymptotic arguments that are unattainable in the nuisance parameter case where the number of parameters grows with the number of observations. Coarse grid numerical integration or Laplace Approximation methods provide answers but do not provide a sense of how far it might be from the actual integral. Currently the only way to assess the quality of the Laplace approximation is to re-fit the model using MCMC, but that requires re-formulating the model into a Bayesian context to ensure ergodicity of the Markov Chain.

This paper explores the use of probabilistic numerical methods for producing diagnostics for the quality of the Laplace Approximation. The probabilistic integration based method evaluates the nuisance parameter integral while characterizing the uncertainty induced by
integrating over a computationally feasible coarse grid. The result is a diagnostic tool for assessing the quality of the Laplace Approximation.

1.1 Laplace Approximation

Consider the function \( f(\theta, t) \) with nuisance parameters \( t \) and structural parameters \( \theta \), where we wish to marginalize over \( t \)

\[
f(\theta) = \int_{T} f(\theta, t) dt,
\]

(1.1)

The Laplace approximation (LA) is used to approximate the integral in (1.1) using a Taylor expansion over \( \log[f(\theta, t)] \) around its maximum \( \hat{t} \) so that the first derivative term is zero,

\[
\log[f(\theta, t)] \approx \log[f(\theta, \hat{t})] - \frac{1}{2} (t - \hat{t})^2 \left(- \frac{d^2}{dt^2} \log[f(\theta, \hat{t})]\right).
\]

(1.2)

Exponentiating both sides of 1.2 and integrating gives the desired integral,

\[
\int_{T} \exp\{\log[f(\theta, t)]\} dt = \int_{T} f(\theta, t) dt
\]

\[
\approx \int_{T} \exp\left\{\log[f(\theta, \hat{t})] - \frac{1}{2} (t - \hat{t})^2 \left(- \frac{d^2}{dt^2} \log[f(\theta, \hat{t})]\right)\right\} dt
\]

\[
= f(\theta, \hat{t}) \int_{T} \exp\left\{- \frac{1}{2} (t - \hat{t})^2 \left(- \frac{d^2}{dt^2} \log[f(\theta, \hat{t})]\right)\right\} dt. \tag{1.3}
\]

The integrand in (1.3) is the kernel of a Normal distribution with an analytic solution,

\[
\int_{T} f(\theta, t) dt \approx f(\theta, \hat{t}) \sqrt{2\pi \left(- \frac{d^2}{dt^2} \log[f(\theta, \hat{t})]\right)^{-1}}. \tag{1.4}
\]

The advantages of using LA are the analytic expression and low computational cost because it only requires calculation of a second derivative [2, 6]. The approximation result is exact if \( f(\theta, t) \) an exponentiated quadratic over \( T \). Deviation from this necessary shape is a common occurrence in nuisance parameter, particularly when in non-linear state space models.

1.2 Motivation

This paper builds a fast diagnostic tool to assess the quality of the LA rather than reformulating the model for Monte Carlo nuisance parameter marginalization. Based on Chkrebtii et al [1], we use a probabilistic integrator over a coarse numerical grid of \( t \) in order to
approximate \( f(\theta, t) \) and its marginal \( f(\theta) \) while characterizing the uncertainty around a coarse resolution integral.
Chapter 2

Methodology

2.1 Probabilistic Solver

In this paper, we define a fast diagnostic tool for assessing the quality of the LA. Although LA is often used to marginalize a likelihood or posterior, for simplicity of terminology we consider the full function \( f(\theta, t) \) and the target function \( f(\theta) = \int_T f(\theta, t) dt \) for some value of \( \theta \).

The proposed probabilistic numerical integration strategy uses a Gaussian Process (GP) prior to model \( f(\theta, t) \) and \( f(\theta) \) as the random variables \([\mu, \mu_t|\theta, \Psi]\), where \( \mu_t \) is the random variable that will be used to model the full function \( f(\theta, t) \), and \( \mu \) models the target function \( f(\theta) \). The vector \( \Psi \) contains hyper-parameters for the Gaussian Process model and \( \theta \) is an auxiliary variable as define in Section 1.1. The GP prior is updated conditional on model interrogations \( f = \{ f(\theta, s) | s = s_1, \ldots, s_c \} \) evaluated over a coarse grid of values of \( s \in T \). The proposed approach can be performed in one shot rather than sequentially over \( T \) as was necessary in [1]. The Bayesian nature of the integration method produces posterior samples from \([\mu | f]\) which can be compared to the value approximated using LA in order to assess the validity of LA for this model. If \( f(\theta) \) is shaped sufficiently like an exponentiated quadratic function with respect to \( t \), then the LA will be consistent with \( \mu | f \). If LA is not a reasonable approximation, alternative, slower marginalization methods should be used instead.

2.2 Probabilistic Solver Prior

Using superscript 0 to represent that 0 additional data has been added to the model, the GP prior on \([\mu^0, \mu_t^0|\theta, \Psi]\), depends on hyper-parameters \( \Psi = [m^0, m_t^0, C^0, C_t^0, \alpha, \lambda] \), where \( m^0, m_t^0 \) are the prior means for \( \mu^0, \mu_t^0 \) respectively with corresponding covariances \( C^0, C_t^0 \) and GP tuning parameters \( \alpha \) and \( \lambda \).
The prior means, $m^0$ and $m^0_t$, are selected under the assumption that the LA is accurate. This implies that $m^0_t$ should be chosen to be shaped as a Gaussian density with location $\hat{t}$, the maximum value of $f(\theta, t)$, and scale equal to $\left(-\frac{d^2}{dx^2}\log[f(\theta, \hat{t})]\right)^{-1}$. The prior $m^0$ is shaped as the corresponding Gaussian density integrated over the interval $(t_1, t_n)$. The GP model is evaluated on the set of points $t \in \mathcal{T}$, where $s \subset t = \{t_1, \ldots, t_n\}$ and $c \leq n$, but evaluation points, $t$, and interrogation points, $s$, have the same lower and upper bounds.

The GP covariance functions $C$ and $C_t$ are determined by the intrinsic correlation function (ICF) used to define the smoothness of $f$ and $\mu_t$ [4, 1]. The squared exponential covariance and the uniform covariance are two popular kinds of ICF (i.e. covariance kernels):

<table>
<thead>
<tr>
<th>Squared Exponential Covariance (Kernel)</th>
<th>$R_\lambda(t_j, t_k) = e^{-(t_j-t_k)^2/2\lambda^2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform Covariance (Kernel)</td>
<td>$R_\lambda(t_j, t_k) = 1_{(t_j-\lambda, t_j+\lambda)}(t_k)$</td>
</tr>
</tbody>
</table>

The squared exponential covariance produces infinitely differentiable smooth functions. However, its unbounded support results in a high computational cost, especially in high dimensions or when the number of estimation points becomes large. The compactly supported uniform covariance function has adequate performance approximating many kinds of continuous functions at lower computational cost.

The covariance functions use $\alpha$ and $\lambda$ to define smoothness over evaluation and interrogation points. The model covariance terms are based on the ICF and the integrated ICF, $Q_\lambda(t_j, t_k) = \int_{t_j}^{t_k} R_\lambda(z, t_k)dz$,

$$C^0_{11} = C(t_j, t_k) = RR(t_j, t_k) = \alpha^{-1} \int_{t_j}^{t_k} R_\lambda(t_j, z) R_\lambda(t_k, z)dz, \quad (2.1)$$

$$C^0_{22} = C(t_j, t_k) = QQ(t_j, t_k) = \alpha^{-1} \int_{t_j}^{t_k} Q_\lambda(t_j, z) Q_\lambda(t_k, z)dz, \quad (2.2)$$

$$C^0_{12} = \int_{t_j}^{t_k} C^0_{11}(t_j, z)dz = RQ(t_j, t_k) = \alpha^{-1} \int_{t_j}^{t_k} R_\lambda(t_j, z) Q_\lambda(t_k, z)dz, \quad (2.3)$$

$$C^0_{21} = \int_{t_j}^{t_k} C^0_{11}(z, t_k)dz = QR(t_j, t_k) = \alpha^{-1} \int_{t_j}^{t_k} Q_\lambda(t_j, z) R_\lambda(t_k, z)dz. \quad (2.4)$$

The resulting GP model is

$$\begin{bmatrix} \mu^0_t(t_j) \\ \mu^0_t(t_k) \end{bmatrix} (t_j, t_k) \sim \mathcal{GP} \left( \begin{bmatrix} m^0_t(t_j) \\ m^0_t(t_k) \end{bmatrix}, \begin{bmatrix} C^0_{11} & C^0_{12} \\ C^0_{21} & C^0_{22} \end{bmatrix} \right). \quad (2.5)$$

The tuning parameter $\lambda$ represents the search range of the evaluation process in units of grid points of $s$. Increasing $\lambda$ increases the effective range over each evaluation action considered, consequently $\lambda$ is usually interpreted as a constant times the distance between two adjacent evaluation points (i.e. $t$). A larger $\lambda$ will help to reduce the bias of evaluation.
in a finer grid. However, if $\lambda$ is too large, it will expand computational burden without improving the accuracy of evaluation.

The second tuning parameter $\alpha$ is the prior precision parameter to control the variance of the evaluated mean surface. A small $\alpha$ introduces large variance on the values of $t \not\in s$. The tuning parameter $\alpha$ is often set to be a proportion of number of total observation points [4, 1]. If $\alpha$ is too large, the variance between each evaluation is small, but the bias may be large. Optimization of tuning parameters is discussed further in Section 2.6.

2.3 Updating the Mean and Covariance

The GP model $[\mu_0, \mu^0_t | \theta, \Psi]$ is updated by conditioning on model interrogations [5] $f = \{ f(\theta, s) \mid s = s_1, \ldots, s_c \}$:

$$
\begin{bmatrix}
\mu^1(t_j) \\
\mu^1(t_k)
\end{bmatrix} \sim \mathcal{GP}\left( m^1(t_j), m^1(t_k) ; 
\begin{bmatrix}
C^1(t_j, t_j) & \int_0^{t_j} C^1(z, t_k)dz \\
\int_0^{t_j} C^1(z, t_k)dz & C^1(t_k, t_k)
\end{bmatrix} \right).
$$

(2.6)

Interrogation points $f$ are evaluated exactly and all in one shot, unlike in [1], where their differential equation model structure requires $f$ to be sequentially predicted ahead and sampled with uncertainty. The uncertainty in $\mu$ is the combination of the prior and $f$ in a mixture defined by the proximity of $t$ to $s$.

The Updates:

1. Update the mean surface of the full and target functions at locations $t_j, t_k \in t$

$$m^1(t_j) = m^0(t_j) + C^0(t_j, s) \left( C^0(s, s) \right)^{-1} (f - m^0_t(s))$$

$$m^1(t_k) = m^0(t_k) + \int_{t_1}^{t_k} C^0(z, s)dz \left( C^0(s, s) \right)^{-1} (f - m^0_t(s))$$

2. Update the covariance surfaces at locations $t_j, t_k \in t$

$$C^1(t_j, t_j) = C^0(t_j, t_j) - \left\{ C^0(t_j, s) \right\} \left( C^0(s, s) \right)^{-1} \left\{ C^0(t_j, s) \right\}^\top$$

$$C^1(t_k, t_k) = C^0(t_k, t_k) - \left\{ \int_{t_1}^{t_k} C^1(z, s)dz \right\} \left( C^0(s, s) \right)^{-1} \left\{ \int_{t_1}^{t_k} C^1(z, s)dz \right\}^\top$$

2.4 The Covariance Kernels

The evaluation result on the same target distribution is different by using different kernels. In Figure 2.1a, the uniform kernel “jumps” to 1 if the distance between two evaluation points is less than $\lambda$. Otherwise, it returns 0. On the other hand, the squared exponential kernel produces a smoother covariance surface as shown in Figure 2.1b.
2.5 Probabilistic Solver in higher Dimensions

The purpose of this paper is to produce a diagnostic tool to assess whether LA is valid for marginalizing a function. In this section, methodology is expanded from one dimension to higher dimensional cases while maintaining a low computational cost for the probabilistic solver.

We first expand the probabilistic solver outlined in Section 2.2 into two dimensions on \( n \) evaluation points, where for clarity of exposition we expand notation of the evaluation set into the two dimensional variable, \((x, t) = [(x_1, t_1), \ldots, (x_n, t_n)] \in X \times T\). Similar to before, \( \mu_{x,t} \) is the full function and \( \mu \) is the target function. The prior of \( \mu_{x,t} \) and \( \mu \) can be built analogous to (2.1)-(2.4) as the product of corresponding kernels and their integrals:

\[
RR([x_j, t_j], [x_k, t_k]) = \alpha^{-1} \int_T \int_X R_\lambda([x_j, t_j], [w, z]) R_\lambda([x_k, t_k], [w, z]) \, dwdz, \tag{2.7}
\]

\[
QQ([x_j, t_j], [x_k, t_k]) = \alpha^{-1} \int_T \int_R Q_\lambda([x_j, t_j], [w, z]) Q_\lambda([x_k, t_k], [w, z]) \, dwdz. \tag{2.8}
\]

Next we expand into \( M \) dimensions \( T = [t_1, \ldots, t_M] \in T_1 \times \cdots \times T_M \) where GP parameter \( \lambda \) is a vector of tuning parameters. Under this scenario, \( \mu \) represents the target function and \( \mu_T \) represent the full function. The uniform kernel is a \( M \) dimensional convolution over points \( Z \in T_1 \times \cdots \times T_M \):

\[
RR(T_j, T_k) = \int_{T_1} \cdots \int_{T_M} \prod_{m=1}^M \alpha^{-1}_m R_\lambda(T_j, Z_m) R_\lambda(T_k, Z_m) dZ_m, \tag{2.9}
\]

\[
QQ(T_j, T_k) = \int_{T_1} \cdots \int_{T_M} \prod_{m=1}^M \alpha^{-1}_m Q_\lambda(T_j, Z_m) Q_\lambda(T_k, Z_m) dZ_m.
\]
2.6 Tuning Parameters

There are two parameters to be chosen before using our method, $\lambda$ and $\alpha$. The parameter $\lambda$ controls the “search range” in both kernels. The value of $\lambda$ represents the tolerance of the kernel estimation. As $\lambda$ increase, the kernels’ surfaces become smoother up to a certain point. The result is highly dependent on the true shape of the target distribution, so the shape of the kernel is not the dominant factor during evaluation. In past studies, points are on a regular grid and the value of $\lambda$ is set to be 1.5 times the distance between neighbouring points [1]. Parameter $\alpha$ is the prior precision. Large $\alpha$ guarantees each sampling result will be similar. The suggested $\alpha$ is equal to the ratio of the scale of inputs and outputs [4] in the one dimensional cases. It is not possible to apply this rule in high dimensional cases.

In this paper, we provide a rule of thumb for $\lambda$ and $\alpha$ based on our optimization experiments. The choice of $\lambda$ and $\alpha$ is essentially a choice between variance and bias. A higher $\lambda$ or a higher $\alpha$ will make the evaluation results of different samples have smaller variation (low variance), but the evaluation results will be systematically far away from the true value (high bias) and vice-versa. In order to choose the proper $\lambda$ and $\alpha$, a genetic optimization algorithm is used to minimize the difference between the approximated function and the real function. Since our goal is to integrate the full function over the interval specified by $t$, we consider quality of approximation based on the estimated integral evaluated at the end point.

There are two risk functions which we use for optimization. The $\mu(\hat{x})$ and $f(\theta|\hat{x})$ represent the estimated value and true value of the target function on the last evaluational point $\hat{x}$ respectively.

$$D_{tuning} = \sum_{i=1}^{B} (\mu_i(\hat{x}) - f(\theta|\hat{x}))^2$$

$$D_{tuningabs} = \sum_{i=1}^{B} |\mu_i(\hat{x}) - f(\theta|\hat{x})|$$

For the optimization experiments we consider, a one-dimensional standard normal distribution (STN) for the purpose of computational convenience and the diagnostics. In the one dimensional case, we can use the tuned $\alpha$ and $\lambda$ directly. In the high dimension case, we use it as a reasonable starting point. In this section, we set the number of interrogation points, $c = 5$, the magnification between observational points and evaluational points, $\omega = 1$, the number of samples at each evaluation point, $B = 2000$.

We use a genetic optimization algorithm [3] and constrain that the optimized point have to the search interval for $(\lambda, \alpha)$ is $(1, 5) \times (1, 100)$. This optimization interval is decided by the two reasons: First, the variation of evaluation result is relatively constant after $\lambda$ and $\alpha$ reach to certain level based on our testing experiment. If the uniform kernel is used,
the evaluation result would not change that much after $\lambda > 5$ in one dimensional cases. Otherwise, if the squared exponential kernel is used, the $C_{11}$ will become singular when $\lambda > 5$. The evaluation result is relatively constant when $\alpha > 50$, we double the boundary range in order to increase the chance that the global minimal is reached. The number of optimization particles used by the optimizer is 2500, and the maximum iterations of \textit{genoud} is equal to 10000. Both risk functions have their pros and cons. The gradient in $D_{tuning}$ function is easier to find, but it amplifies the error when the evaluational result is far away from the true result. On the other hand, the $D_{tuningabs}$ is a more robust risk function. However, it is hard to find the derivative on some combinations of $\alpha$ and $\lambda_c$.

When the uniform kernel is used, the optimization results on $D_{tuning}$ are shown below:

![Graph](image)

(a) The uniform kernel is used when optimized $D_{tuning}$.

(b) The squared exponential kernel is used when optimized $D_{tuning}$.

Figure 2.2: When the target distribution is an one dimensional standard normal distribution, $\alpha \in [1 : 100], \lambda_c \in [1 : 5]$, the behaviour of $D_{tuning}$

The behaviour of $D_{tuning}$ is shown in Figure 2.2. If uniform kernel is used, the $D_{tuning}$ increases when $\alpha$ is small, which is shown in Figure 2.2a. If squared exponential kernel is used, the $D_{tuning}$ has similar behaviour, but when $\alpha$ is small, the value of $D_{tuning}$ is smaller if $\lambda$ is smaller. The uniform kernel tends to have worse evaluational result than the squared exponential kernel, because the full function is a standard normal distribution.

When $D_{tuningabs}$ is the risk function, the optimization result on the Figure 2.3 has similar shape compared to $D_{tuning}$, but the range of the error is larger than the range in $D_{tuning}$ when squared exponential kernel is used. Also, it is easier to see the changes of the fluctuation of the difference, especially when uniform kernel is used in Figure 2.3a. When $D_{tuningabs}$ is the risk function, the tuning result is shown in table 2.1. The optimal $\lambda_c$ is 1.000419 if squared exponential kernel is used, which is similar to Chkrebtii[1] suggest in her paper, but the choice of $\alpha$ is much larger compared to Skilling and Chkrebtii used[4][1].

9
(a) The uniform kernel is used when optimized $D_{tuningabs}$.

(b) The squared exponential kernel is used when optimized $D_{tuningabs}$.

Figure 2.3: When the target distribution is an one dimensional standard normal distribution, $\alpha \in [1 : 100]$, $\lambda_c \in [1 : 5]$, the behaviour of $D_{tuningabs}$.

Table 2.1: The tuning result of $\alpha$ and $\lambda_c$ ($\hat{\alpha}$ and $\hat{\lambda}_c$) in one dimensional Cases

<table>
<thead>
<tr>
<th></th>
<th>Uniform Kernel</th>
<th>Squared Exponential Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\alpha}$</td>
<td>99.126495</td>
<td>64.710878</td>
</tr>
<tr>
<td>$\hat{\lambda}_c$</td>
<td>1.000419</td>
<td>3.687494</td>
</tr>
<tr>
<td>$D_{tuningabs}$</td>
<td>100</td>
<td>65.077103</td>
</tr>
<tr>
<td>$D_{tuning}$</td>
<td>1</td>
<td>2.913376</td>
</tr>
</tbody>
</table>

It is doubtful that genoud method converges to the global minimum, because the optimization process is stopped when the maximum number of iterations is hit, and the reason is at least one gradient is larger than the tolerance, $10^{-3}$. The gradient, which is closest to 0, is called the optimal gradient in this paper. The optimal gradient, when the uniform kernel is used, is $-2.921819$. The optimal gradient, when the squared exponential kernel is used, is $7.222098 \times 10^{-3}$. This is also evidence to show the evaluation result will have more variance but less bias if uniform kernel is used, and the result will have more bias but less variance if the squared exponential kernel is used. This means the global minimum would be somewhere else within the boundary area or even outside of boundary. However, the value of $D_{tuning}$, which is on the chosen $\alpha$ and $\lambda_c$, within the global minimal area. Although the global minimum of $D_{tuning}$ may not be reached, the chosen $\alpha$ and $\lambda_c$ is a reasonable starting point to provide a low $D_{tuning}$.

We tried 1000, 5000, and 10000 as the maximal number of iterations, the genoud method cannot find the global minimum and the optimal $D_{tuning}$ is similar under all three scenarios. We choose a maximal number of iterations of 1000 in this paper for computational convenience.
2.7 Simple Experiments

Our method provides the distribution of the target function at the upper boundary of the interval. If the LA produces a value consistent with the density, then LA is considered a valid method, otherwise LA is not reasonable and alternative methods should be used. If LA gives a value below the distribution of probabilistic integrals, the full function has wider tails than would be expected from an exponentiated quadratic.

For the convenience of comparing and computational cost, we set the number of observations points \( c \) is equal to 5; the magnification between observational points and evaluational points (i.e. \( \omega \)) is equal to 10; the number of sample \( B \) is equal to 50; the tuning parameters \( \alpha \) and \( \lambda_c \), were set as per the optimization results above. The target distribution is a one dimensional (positively skewed) Gamma distribution with shape parameter of 2 and rate parameter of 4.

In this section we compare the evaluation results and computational costs between the squared exponential kernel and the uniform kernel [4][1]. In the one dimensional case, the computational time for both kernel are around 6.5 seconds. In the two dimensional case, the computational time of the squared exponential kernel is around 2800 seconds, and it is around 2200 seconds for the uniform kernel. This is an evidence that uniform kernel has lower computational cost. In addition, the evaluational results are shown in Figure 2.4.
The variance and bias trade off not only exists in the choice of \( \lambda_c \) and \( \alpha \) but also in the kernel selection. Here the \( \hat{\lambda}_c \) and \( \hat{\alpha} \), which are different for each kernel, are chosen for minimize the \( D_{tuning} \). The full function is non-stationary in that it changes rapidly on the left and the high skew produces slower changes on the upper tail. The uniform kernel “learns” the convex shape around the mode and applies it to the next evaluation action. The squared exponential kernel does an excellent job on the left side of the target distribution. However, the kernel assumes a stationary and symmetric function. The squared exponential kernel “thinks” the full function between the fourth and fifth observational point will have a similar shape as the last step and it does not want to underestimate it again, so it tends to change its shape and “give” a higher evaluation result. Nevertheless, the tail part of the true function is likely linear, so the evaluation result is way off at this part. The evaluation result of each sample is almost the same during the whole evaluational process.

The different properties of the two kernels also have an impact on the evaluation of the target function. The uniform kernel does not mimic the full function well on the left side of the mode, so this evaluation bias causes target function underestimate the true target function.
function. In Figure 2.4b, the curve of the evaluated target function is smaller than the true target function on the mode (i.e., third observational point), and the kernel tries to “fix” the problem by overestimating the target function of the distribution between third and fourth observational point, so the evaluated target function reach to 1 before the fourth observational point. On the other hand, the squared exponential kernel mimics the target function better than the uniform kernel, the estimated target function is the same as the true target function on the third observational point, but it underestimated the target function on between the third and fourth observational point, so it overestimates the target function between the fourth and fifth observational point. The evaluated target function reaches to 1 after the fourth observational point but has a similar evaluated result on the fifth point, which is shown in Figure 2.4e.

The result of the bias and variance tradeoff between kernels is highlighted in figures 2.4c and 2.4f where the integral result from the uniform kernel has larger variance but smaller bias compared the squared exponential kernel when applied to a skewed full function.

2.7.1 Choice of Evaluation Grid

With an increasing size for $c$, we explore two ways of choosing $(s)$: “standard deviation extend” method and “fill in” method. In “standard deviation extend”, the total evaluation range is extended when adding more points. The “fill in” method uses a fixed range for $s$ and adds more information by shortening the distance between points.

The uniform kernel is used for saving computational time. The full function is still a positively skewed one dimensional gamma distribution. All settings are the same as in Section 2.7 except the number of observational points increases to 7.
Figure 2.5: The first row represents the evaluated results with the true values of the full function, the target function, and the histogram of evaluated target function on the last evaluation point when the “fill in” method is used. The second row represents the corresponding evaluated results with the true values when the “standard deviation extend” method is used, and the vertical line is the LA value.

The results of changing $c$ is shown in Figure 2.6 based on the “standard deviation extend method”. As seen in Figures 2.4a and 2.5d, the problem of overestimation between third and fourth observational points is not solved, compare to the evaluational result when there is 5 observational points. The evaluational accuracy is increased, because the mode and the mean of the histogram in Figure 2.5c is on the left side of 1.20, and the mode and the mean of the histogram in Figure 2.4c is on the right side of 1.20. In other words, when $c$ equals to 5, it has already detected the difference between the target distribution and evaluational result well. On the other hand, the “standard deviation extend” method increases the search range by one standard deviation on each side, but the evaluational result is worse when the number of observational points is larger.

When “standard deviation extend” method is used (i.e. Figure 2.6), the estimation bias increase when the number observational points increase. The more observational points means more of the tail part of the target distribution is evaluated, and there are extra estimation bias on the extended tail part of the target distribution. On the other hand,
Figure 2.6: The histogram on the last evaluational point when “standard deviation extend” method for $c = 5$, 7, 9, 11, 15, and 17 respectively, the vertical line is the LA value.
when “fill in” method is used (i.e. Figure 2.7), the estimation variance decrease when the number of observational points increase. The estimation range of the target distribution is not change, but more observational points give a more precise estimation on the fix range. For the remainder of the paper, the number of observational points is set to 5 to maintain a low computational requirement while maintaining a decent job on detecting the difference between the evaluational result and target distribution. When the number of observational points is set to 5, there are no difference related to the method used.

Figure 2.7: The histogram on the last evaluational point when “fill in” method for $c = 5, 7, 9, 11, 15,$ and $17$ respectively, and the vertical line is the LA value.
Chapter 3

Applications

We apply our normality check method to a two dimensional cases. In this chapter, we test our method on three multivariate distributions. The test range interval is two standard deviations on each side of the mean based on the assumption that the full function is shaped as a Normal distribution. This range corresponds to an integral of the target function of 0.9545 if LA is exact. The number of evaluation points in $t$ and the number of samples are both set to 50 in two dimensional cases due to the limits of computational power and API usage of the library $ploty$. The uniform kernel is used in this case for the sake of time saving and lower bias in high dimensional case.

3.1 One Dimensional Normal Distribution

In this subsection, the target distribution is a one dimension normal distribution with mean equal to 2.6 and standard deviation equal to 9.327118. The $\alpha$ and $\lambda_c$ equal to 99.93421 and 1.00554 respectively.

The estimation result of the full function and the target function of the target distribution as shown in Figure 2.4. Comparing Figure 3.1a to Figure 3.1d and Figure 3.1b to Figure 3.1e, the estimation result of the full function and the target function on the estimation points, which are not observational points, has larger variance when the uniform kernel is used. On the other hand, the bias of estimations of these two kernels are similar. In this special case, the squared exponential kernel gives a perfect estimation result, because it is similar to the normal distribution.

3.2 Bivariate Normal Distribution

In this subsection, we apply our method on a bivariate normal distribution with mean vector equal to $\begin{bmatrix} 2.6 \\ 2.6 \end{bmatrix}$ and variance covariance matrix equal to $\begin{bmatrix} 2.61 & 1.68 \\ 1.68 & 8.91 \end{bmatrix}$. The $\alpha$ and $\lambda_c$
Figure 3.1: The first row represents the evaluated results and true values of the full function, the target function, and the histogram of evaluated target function on the last evaluation point when the uniform kernel is used. The second row represents the corresponding evaluated results with true values when the squared exponential kernel is used, and the vertical line is the LA value.
equal to 99.93421 and 1.00554 respectively.

Figure 3.2

Figure 3.2 shows the estimated result of the full function of target distribution. This shows that the square exponential kernel (i.e. Figure 3.2a) cause much smaller variance than uniform kernel (i.e. Figure 3.2b) during estimation. On the other hand, the squared exponential kernel tends to underestimated the target function after a certain point.

Figure 3.3

The estimation of target function shows a similar result. The Figure 3.3b shows that the square exponential kernel causes a smaller variance compare to the uniform kernel (i.e Figure 3.3a), and this result is also shown on the histogram of evaluation on the last point of target function. Although the square exponential kernel tends to slightly underestimated
the target function, which is shown in Figure 3.4. However, it is hard to conclude a variance and bias trade off here.

![Figure 3.4](image)

(a) The histogram of evaluating target function on the last estimation point by using uniform kernel, and the vertical line is the LA value.

(b) The histogram of evaluating target function on the last estimation point by using square exponential kernel, and the vertical line is the LA value.

The square exponential kernel tends to give a better approximation result when the full function is a multivariate normal distribution, because it has much smaller variance and negligible bias compared to the uniform kernel.

### 3.3 T Distribution

The T distribution behaviour is asymptotically equivalent to a normal distribution when the degrees of freedom, \( \nu \), becomes larger. However, when \( \nu \) is small, the T distribution has a fatter tail. In this example, we choose a T distribution shifted and scaled to have same value of mean and variance parameters of the normal distribution in Section 3.1, and \( \nu \) equals to 5 making it an approximation to the normal distribution used in Section 3.1 in order to assess if the algorithm can assess differences in target function for these distributions.

#### 3.3.1 One Dimensional T Distribution

The variance bias trade off is shown in Figure 3.5. The estimation results of full function and target function have larger variance when uniform kernel is used. In addition, the square exponential kernel tends to underestimate the target function when we compare Figure 3.5b
to Figure 3.5e. The estimation result of the target function by the square exponential kernel is usually smaller than the true value, but the estimation result of the target function by the uniform kernel is approximately evenly distributed on both sides of the true value. This conclusion is also supported by the histogram of the evaluation result of target function on the last evaluational point, the peak of the histogram, which square exponential kernel is used, in Figure 3.5c is on the left of the true value. On the other hand, the peak of the histogram, in which the uniform kernel is used, in Figure 3.5f is evenly distributed on both sides of the true value.

Figure 3.5: The first row represents the evaluated results and the true values of the full function, the target function, and the histogram of evaluated target function on the last evaluation point when the uniform kernel is used. The second row represents the corresponding evaluated results with the true values when the squared exponential kernel is used, and the vertical line is the LA value.

3.3.2 Two Dimensional T distribution

The target multivariate T distribution is a non-perfect approximation with \( \nu \) equals to 5 of a bivariate normal distribution with mean vector equal to \( \begin{bmatrix} 2.6 \\ 2.6 \end{bmatrix} \), and scale matrix equal to \( \begin{bmatrix} 2.61 & 1.68 \\ 1.68 & 8.91 \end{bmatrix} \). The variance-covariance matrix equals to scale matrix times \( \frac{\nu}{\nu - 2} \)
Figure 3.6 shows the estimated result of target distribution’s full function. This shows the square exponential kernel (i.e. Figure 3.6a) causes smaller variance than the uniform kernel (i.e. Figure 3.6b) during estimation.

The estimation of target function shows similar result. Figure 3.7b shows the square exponential kernel causes a smaller estimation variance compared to the uniform kernel (i.e Figure 3.7a). On the other hand, the square exponential kernel tends to underestimated the target function. This result is also shown on the histogram of evaluation on the last point of target function in Figure 3.8.
3.3.3 Conclusion

The multivariate t distribution is a non-perfect approximation of a related multivariate normal distribution when \( \nu \) is small. However, our algorithm is capable to detect the difference of the distributions which have similar but not identical densities.

If the multivariate t distribution’s mean vector and scale matrix are the same as a multivariate normal distribution’s mean vector and variance-covariance matrix, the estimation result of the multivariate normal distribution’s target function has smaller estimation variance compared to the multivariate T distribution in both the one dimensional and two dimensional cases. This means the estimation on the target function of the target distribution will have a larger variance when it is farther from normality.

3.4 Gamma Distribution

The gamma distribution behave differently than normal distribution, because its support is \([0, \infty)\) and its shape can be non-symmetric. In this section, we choose a gamma function that is highly positively skewed and we expect our estimation result will be far off from the true result. The values of the shape and rate parameters are equal to 2 and 4 respectively.
3.4.1 One Dimensional Gamma Distribution

As it is shown in Figure 3.9, the estimation results of the full function and the target function have larger estimation variance when the uniform kernel is used. In addition, both kernels tend to overestimate the target function, but the uniform kernel gives a closer estimation to the true value compared to the square exponential kernel when we compare Figure 3.9b to Figure 3.9e, because the mode of the histogram is close to 1. We will discuss this in more detail in Section 3.5.

![Figure 3.9](image)

(a) (b) (c)

(d) (e) (f)

Figure 3.9: The first row represents the evaluated results and true values of the full function, the target function, and the histogram of evaluated target function on the last evaluation point when the uniform kernel is used. The second row represents the corresponding evaluated results with true values when the squared exponential kernel is used, and the vertical line is the LA value.

3.4.2 Two Dimensional Gamma Distribution

The target two dimensional gamma distribution is created by two independent gamma distribution with values of the shape and the rate parameters equal to 2 and 4 respectively. It is a highly skewed distribution, and we expected the estimation of the target function is bad.
Figure 3.10

Figure 3.6 shows the estimated result of the target distribution’s full function. This shows the square exponential kernel (i.e. Figure 3.10a) causes larger estimation bias than uniform kernel in the tail (i.e. Figure 3.10b).

Figure 3.11

Because the number of samples, $B$, is small, the estimation of the target at the final point shows similar results. Figure 3.11b shows the square exponential kernel provides almost identical results to the uniform kernel (i.e. Figure 3.11a). Moreover, the square exponential kernel tends to slightly overestimate the target function, which is shown in Figure 3.12.
3.5 Estimation performance on different distributions

In this section, we use $B = 10^5$ samples in order to test the performance of each kernel and examine the density via a histogram of the last evaluation point. The range of the histograms’ plots is set from 0.5 to 1.5 when target distribution is either the normal distribution or the T distribution, and we set the range of the histograms’ plots from 0.5 to 1.5 when the target distribution is gamma distribution.
Figure 3.13: The first row represents the evaluated target function of normal, T, and gamma distribution with the uniform kernel respectively, and the second row represents the evaluated target functions of normal, T, and gamma distribution with the square exponential kernel respectively, and the vertical line is the LA value.

In the one dimensional cases, the uniform kernel tends to slightly underestimate the target function when the target distribution is highly normal, but when the target distribution is highly positively skewed, it tends to overestimate the target function and is far away from the true target function.

On the other hand, the squared exponential kernel tends to give a better estimation when the target distribution is normal, because it has lower estimation and bias variance compared to uniform kernel. Moreover, it shows same trends as the uniform kernel, which is the overestimation will happened when the target distribution is highly positively skewed, and underestimation will happened when the target distribution is similar to normal but not normal. In addition, the squared exponential kernel have smaller estimation variance and larger estimation bias compared to the estimation result by using the uniform kernel when the target distribution is normal likely.
Figure 3.14: The first row represents the evaluated target functions of normal, T, and gamma distribution with uniform kernel respectively, and the second row represents the evaluated target functions of normal, T, and gamma distribution with square exponential kernel respectively, and the vertical line is the LA value.

In the two dimensional cases, both kernels have similar behaviour as they did in one dimension, but the scale of overestimation increases. In other words, the estimation’s bias increases when the dimension of the target distribution increases, when the full function is highly positively skewed. The estimated target function is mainly located on the left side of 1.2 in one dimensional cases, but it is located mainly on the right side of 1.2 in the two dimensional cases. This is shown in the comparison of Figure 3.14c and Figure 3.14f.
In addition, the variance and bias trade off is more obvious when the full function is normal or similar to normal. The estimations of the target function by using the uniform kernel on the last evaluation point have larger variances, but they are evenly distributed on both sides.
Moreover, both kernels tend to have higher estimation variance when the dimensionality of target distribution is higher and the target distribution is similar to normal, because we can tell the range of the histogram of estimating the target function of bivariate normal distribution (Figure 3.14a vs Figure 3.13a) and the two dimensional T distribution (Figure 3.14b vs Figure 3.13b) is larger compared to their one dimensional cases respectively.
In conclusion, the uniform kernel has small estimation bias and larger variance in the two dimensional cases. Compared to one dimensional cases, the estimation results have larger
bias and variance on the bivariate normal distribution and two dimensional T distribution, but they have larger bias and smaller variance on the multivariate gamma distribution.
Chapter 4

Discussion

4.1 Summary

In this paper, we define a new algorithm to test the appropriateness of the LA. This is equivalent to assessing the normality of a target distribution. We use a Gaussian process prior on the joint distribution of target and its integral and produce a posterior distribution in a single step update.

Before we start the evaluation process, we need to decide the way to select evaluation points, the number of observational points, and values of two tuning parameters, \( \lambda_c \) and \( \alpha \). We have two different ways to distribute observational points, and we decided to use the “fill-in” method, because additional efforts focused on the tails of the target do not increase the estimation accuracy that much. In addition, we find 5 observational points are good enough for one and two dimensional cases, because the test show it is enough to detect differences between the estimation result and Laplace Approximation. Using few points saves computational time and will make it possible to extend this work into higher dimensions.

We chose two different risk functions to find the optimal values of \( \lambda_c \) and \( \alpha \) when the target function is shaped like a normal distribution. Optimization in higher dimensional cases was left for future work because of the high computational cost.

Both the uniform kernel and the squared exponential kernel can be used for estimation. The uniform kernel tends to have better performance on the tail of the target distribution with lower computation cost in general, but the squared exponential kernel does a better job on smooth functions. We highlight the trade-off between kernels as a variance, bias trade-off based on our goal of estimating the integral of the full function. If the target distribution is normal distribution, the square exponential kernel performs uniformly better compare to uniform kernel, because of its structure is smooth, differentiable, and stationary. The uniform kernel has a larger estimation variance, but this does not mean uniform kernel is bad, because the true integral value is still included in the histogram of evaluated target function on the last evaluational point. More generally, we found that the uniform kernel
has a larger estimation variance but smaller estimation bias. The square exponential kernel underestimated the target function when it is symmetric but having fat tails. As a result the true value of target function is not within the histogram of posterior samples. When the number of dimension of the target distribution increases, the evaluation result changes as well. When a non-normal target distribution has higher dimension, the estimation result will have higher estimation bias but smaller estimation variance. On the other hand, the estimation bias is stable when target distribution is normal likely, but the estimation variance increase visibly.
Bibliography


Appendix A

Construct the standard deviation surface

**Standard Deviation Surface(\(\hat{S}\)):**

1. Choose the corresponding variance matrix \(C\) and transform to symmetric matrix \(C^*\)
   \[ C^* = \frac{C + C^\top}{2} \]
2. Find the eigenvalue \(\lambda\) and eigenvector \(\nu\) of \(C^*\)
3. Making sure all eigenvalues are positive
   \[ \lambda^* = \begin{cases} 
   \lambda & \text{if } \lambda \geq 0 \\
   \lambda + |\min(\lambda)| & \text{if } \lambda < 0 
   \end{cases} \]
4. Transform \(\lambda^*\) into a diagonal matrix \(\lambda^{**}\)
5. Compute the standard deviation matrix
   \[ \sqrt{\Sigma} = \sqrt{\nu \lambda^{**} \nu^\top} = \nu(\lambda^{**})^{\frac{1}{2}} \]
6. Compute the standard deviation surface by sampling
   \[ \hat{S} = (\nu \lambda^{**})^{\frac{1}{2}} \mathcal{N}(0, I) \]
7. Repeat the sampling procedure (i.e. step 6) \(B\) times
Appendix B

Uniform kernel in one and high dimensions

B.1 R: Intrinsic correlation function (ICF)

B.1.1 1-D case

\[ R_\lambda(t_j, t_k) = 1_{(t_j - \lambda, t_j + \lambda)}(t_k) \]

B.1.2 2-D case

\[ R_{\lambda_x, \lambda_t}([x_j, t_j], [x_k, t_k]) = 1_{(x_j - \lambda_x, x_j + \lambda_x)}(x_k) 1_{(t_j - \lambda_t, t_j + \lambda_t)}(t_k) \]

B.1.3 N-D case

\[ R_\lambda(x_j, x_k) = \prod_{i=1}^{N} 1_{(x_{j(i)} - \lambda_{(i)}, x_{j(i)} + \lambda_{(i)})}(x_{k(i)}) \]

The product indicator is taken over the \( i^{th} \) row of the column vectors \( x_j, i = \{1, \ldots, N\} \) and \( \lambda_{(i)} \), where \( i \) denotes the dimension of the system.

B.2 Q: Integrated ICF

B.2.1 a to \( t_j \)

1-D case

\[
Q_\lambda(t_j, t_k) = \int_{t_j}^{t_k} R_\lambda(z, t_k) \, dz \\
= \int_{t_j}^{t_k} 1_{(z - \lambda, z + \lambda)}(t_k) \, dz \\
= \left[ \max\{t_j, t_k + \lambda\} - \min\{a, t_k - \lambda\} \right] 1_{(0, \infty)} \left[ \max\{t_j, t_k + \lambda\} - \min\{a, t_k - \lambda\} \right]
\]
2-D case

\[ Q_\lambda([x_j, t_j]; [x_k, t_k]) = \int_{x_j}^{x_k} \int_{t_j}^{t_k} R_{\lambda_x, \lambda_t}([x_j, t_j], [w, z])dw \, dz \]

\[ = \int_{x_j}^{x_k} \int_{t_j}^{t_k} 1_{(\lambda_x, \lambda_t)}([x_j, t_j], [w, z])dw \, dz \]

\[ = \min\{x_j, x_k + \lambda_x\} - \max\{a_x, x_k - \lambda_x\} \times \]

\[ \min\{t_j, t_k + \lambda_t\} - \max\{a_t, t_k - \lambda_t\} \times \]

\[ \lambda (0, \infty) \{\min\{x_j, x_k + \lambda_x\} - \max\{a_x, x_k - \lambda_x\}\} \times \]

\[ \lambda (0, \infty) \{\min\{t_j, t_k + \lambda_t\} - \max\{a_t, t_k - \lambda_t\}\} \]
B.3.3 N-D case

\[
RR_x(x_j, x_k) = \prod_{i=1}^{N} \alpha_i^{-1} \left[ \min(x_j, i), x_k, i) - \max(x_j, i), x_k, i) + 2\lambda(i) \right] \times \left[ \min(x_j, i), x_k, i) - \max(x_j, i), x_k, i) + 2\lambda(i) \right]
\]

The \( \alpha \) is a length \( N \) vector, and \( \lambda(i) \) is the \( i \)th row of it.

B.4 QR: The covariance of \( \mu_j^0 \) and \( \mu_i^0 \)

B.4.1 a to \( t_j \)

1-D case

\[
QR(t_j, t_k) = a_j^{-1} \int_0^t Q_x(t_j, z) R_x(t_k, z) dz
\]

\[
= a_j^{-1} \left\{ \left( t_j + \lambda \right) \left( \min(t_k + \lambda, t_j + \lambda) - \max(t_k - \lambda, t_j - \lambda + a) \right) - \frac{1}{2} \left[ \min(t_k + \lambda, t_j + \lambda)^2 - \max(t_k - \lambda, t_j - \lambda + a)^2 \right] \right\}
\]

\[
+ \left( t_j - a \right) \left( \min(t_k + \lambda, t_j + \lambda) - \max(t_k - \lambda, t_j - \lambda) \right)
\]

\[
+ \frac{1}{2} \left( \min(t_k + \lambda, t_j + \lambda) + (t_k - \lambda)^2 \right)
\]

\[
+ \frac{1}{2} \left( \min(t_k + \lambda, t_j - \lambda, t_j + \lambda + a) - t_k + \lambda \right)
\]

2-D case

\[
QR([x_j, t_j], [x_k, t_k]) = a_x^{-1} a_j^{-1} \int_0^t \int_0^t \int_0^t Q_x([x_j, t_j], [w, z]) R_x([x_k, t_k], [w, z]) dwdx dz
\]

\[
= a_x^{-1} a_j^{-1} \left\{ \left( [x_j + \lambda] \left( \min([x_k + \lambda, t_j + \lambda]) - \max([x_k - \lambda, t_j - \lambda + a]) \right) - \frac{1}{2} \left[ \min([x_k + \lambda, t_j + \lambda]^2 - \max([x_k - \lambda, t_j - \lambda + a]^2) \right] \right) \right\}
\]

\[
+ \left( t_j - a \right) \left( \min([x_k + \lambda, t_j + \lambda] - \max([x_k - \lambda, t_j - \lambda + a]) \right)
\]

\[
+ \frac{1}{2} \left( \min([x_k + \lambda, t_j + \lambda] + (t_k - \lambda)^2 \right)
\]

\[
+ \frac{1}{2} \left( \min([x_k + \lambda, t_j - \lambda, t_j + \lambda + a] - t_k + \lambda \right)
\]

\[
\times \left\{ \left( [x_j + \lambda] \left( \min([x_k + \lambda, x_j + \lambda] - \max([x_k - \lambda, x_j - \lambda + a]) - \frac{1}{2} \left[ \min([x_k + \lambda, x_j + \lambda]^2 - \max([x_k - \lambda, x_j - \lambda + a]^2) \right] \right) \right\}
\]

\[
+ \left( x_j - a \right) \left( \min([x_k + \lambda, x_j + \lambda] - \max([x_k - \lambda, x_j - \lambda]) \right)
\]

36
N-D case

\[ QR_k(x_j, x_k) = \prod_{i=1}^{N} \alpha_{(i)}^{-1} \left\{ \left[ (x_{j,(i)} + \lambda_{(i)}(\min[x_{k,(i)} + \lambda_{(i)}, x_{j,(i)} + \lambda_{(i)}) - \max[x_{k,(i)} - \lambda_{(i)}, x_{j,(i)} + \lambda_{(i)}]) + \right. \right. \notag \\
\left. \left. \frac{1}{2}(\min[x_{k,(i)} + \lambda_{(i)}, x_{j,(i)} + \lambda_{(i)}) - \max[x_{k,(i)} - \lambda_{(i)}, x_{j,(i)} - \lambda_{(i)}]) + \right. \right. \notag \\
\left. \left. \max[x_{k,(i)} - \lambda_{(i)}, x_{j,(i)} + \lambda_{(i)}]) - \max[x_{k,(i)} + \lambda_{(i)}, x_{j,(i)} - \lambda_{(i)}]) + \right. \right. \notag \\
\left. \left. (x_{j,(i)} - a)(\min[x_{k,(i)} + \lambda_{(i)}, x_{j,(i)} - \lambda_{(i)}]) - \max[x_{k,(i)} - \lambda_{(i)}, x_{j,(i)} - \lambda_{(i)}]) + \right. \right. \notag \\
\left. \left. 10_\infty \{\min[x_{k,(i)} + \lambda_{(i)}, x_{j,(i)}] - \lambda_{(i)}\} - \max[x_{k,(i)} - \lambda_{(i)}, x_{j,(i)}] + \lambda_{(i)}) \right\} \right\} \notag \\
\notag \\
\notag \notag \\
\notag \notag \\
\notag \notag \\

Indefinite Integral

1-D case

\[ QR(t_j, t_k) = \alpha_t^{-1} \int_{\mathbb{R}} Q_{\lambda}(t_j, z) R_{\lambda}(t_k, z) dz \] 
\[ = \alpha_t^{-1} \int_{\mathbb{R}} 2\lambda (t_k - \lambda, t_j + \lambda) z dz \] 
\[ = \alpha_t^{-1} \int_{\mathbb{R}} 4\lambda^2 z dz \]

2-D case

\[ QR([x_j, t_j], [x_k, t_k]) = \alpha_t^{-1} \alpha_x^{-1} \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} Q_{\lambda}([x_j, t_j], [w, z]) R_{\lambda}([x_k, t_k], [w, z]) dw dz \] 
\[ = \alpha_t^{-1} \alpha_x^{-1} \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} 16\lambda^4 dw dz \]

N-D case

\[ QR_{\lambda}(x_j, x_k) = \prod_{i=1}^{N} \alpha_{(i)}^{-1} 4\lambda^2 \]
B.5 QQ: The variance of $\mu^0$

B.5.1 a to $t_j$

1-D case

$$QQ_\lambda(t_j, t_k) = \alpha^{-1} \int_{\mathbb{R}} Q_\lambda(t_j, x)Q_\lambda(t_k, x)dx$$

$$= \alpha^{-1} \left\{ (t_j - a)(t_k - a)(a + 2\lambda - \max\{t_k, t_j\}) ight.$$

$$+ \left. \frac{1}{2}(\min(t_k + \lambda, t_j + \lambda)^3 - \max\{t_k - \lambda, t_j - \lambda, a + \lambda\}^3) - \frac{1}{2}(t_k + t_j + 2\lambda)(\min(t_k + \lambda, t_j + \lambda)^2 - \max\{t_k - \lambda, t_j - \lambda, a + \lambda\}^2) \right.$$

$$\times (\lambda^2 + t_k\lambda + t_j\lambda + t_kt_j)(\min(t_k + \lambda, t_j + \lambda) - \max\{t_k - \lambda, t_j - \lambda, a + \lambda\}) \right.$$

$$+ \left. \frac{1}{2}(t_j - a)(\min(t_k - \lambda, a + \lambda)^2 - (t_j - \lambda)^2) + (t_j - a)(\lambda - a)(\min(t_k - \lambda, a + \lambda) - t_j + \lambda) \right.$$

$$+ \left. \frac{1}{2}(t_k - a)(\min(t_j - \lambda, a + \lambda)^2 - (t_k - \lambda)^2) + (t_k - a)(\min(t_j - \lambda, a + \lambda) - t_k + \lambda) \right.$$

$$+ \left. \frac{1}{2}(-\lambda^2 - t_k\lambda - t_j\lambda - t_kt_j)(\min(t_k + \lambda, t_j + \lambda) - \max\{t_k - \lambda, t_j - \lambda, a + \lambda\}) \right.$$

$$+ \left. \frac{1}{3}(\min(t_k - \lambda, t_j - \lambda, a + \lambda)^3 - (a - \lambda)^3) + (\lambda - a)(\min(t_k - \lambda, t_j - \lambda, a + \lambda)^2 - (a - \lambda)^2) + (\lambda - a)^2(\min(t_k - \lambda, t_j - \lambda, a + \lambda) - a + \lambda) \right.$$

$$+ \left. 4\lambda^2[\min\{t_k - \lambda, t_j - \lambda\} - \lambda - a] \right\} \left\{ \right.$$
\[ \mathbb{E}_{\beta}^n(\mathbf{x}) = \frac{1}{n^2} \sum_{i,j=1}^{n} \mathbb{E}_{\beta}^{ij}(\mathbf{x}) \]

where \( \mathbb{E}_{\beta}^{ij}(\mathbf{x}) \) is defined as:

\[ \mathbb{E}_{\beta}^{ij}(\mathbf{x}) = \frac{1}{\beta} \exp\left( \frac{-\beta}{2} \sum_{k=1}^{n} \left( x_k - \mathbf{x}_k \right)^2 \right) \]

for \( \beta > 0 \) and \( \mathbf{x} \in \mathbb{R}^n \), with \( \mathbf{x}_k \) being the \( k \)-th component of \( \mathbf{x} \).
(λ_1^2 + x_k\lambda_k + x_j\lambda_x + x_jx_k)(\min\{x_k + \lambda_x, x_j + \lambda_x\} - \max\{x_k - \lambda_x, x_j - \lambda_x, a_x + \lambda_x\})
+ \frac{1}{2}(x_j - a_x)(\min\{x_k - \lambda_x, a_x + \lambda_x\})^2 - (x_j - \lambda_x)^2 +
(x_j - a_x)(\lambda_x - a_x)(\min\{x_k - \lambda_x, a_x + \lambda_x\} - x_j + \lambda_x)
+ \frac{1}{2}(\min\{x_k - \lambda_x, x_j + \lambda_x\})^2 - \max\{x_j - \lambda_x, a_x + \lambda_x\})^2
+ \frac{1}{2}(\min\{x_k - \lambda_x, x_j + \lambda_x\} - x_j + \lambda_x)
+ 2\lambda_x[(x_j + \lambda_x)(\min\{x_k - \lambda_x, x_j + \lambda_x\} - \max\{x_j - \lambda_x, a_x + \lambda_x\}) -
\frac{1}{2}(\min\{x_k - \lambda_x, x_j + \lambda_x\})^2 - \max\{x_j - \lambda_x, a_x + \lambda_x\})^2
+ (x_k + \lambda_x)(\min\{x_k + \lambda_x, x_j - \lambda_x\} - \max\{x_k - \lambda_x, a_x + \lambda_x\})
+ \frac{1}{3}(\min\{x_k - \lambda_x, x_j - \lambda_x, a_x + \lambda_x\})^3 - (a_x - \lambda_xa)^3 +
(\lambda - a_x)(\min\{x_k - \lambda_x, x_j - \lambda_x, a_x + \lambda_x\})^2 - (a_x - \lambda_x)^2 +
(\lambda - a_x)^2(\min\{x_k - \lambda_x, x_j - \lambda_x, a_x + \lambda - a_x + \lambda\})
+ \frac{1}{2}(\min\{x_k - \lambda_x, x_j - \lambda_x, a_x + \lambda_x\} - a_x + \lambda_x)
+ 4\lambda_x^2(\min\{x_k - \lambda_x, x_j - \lambda_x\} - \lambda_x - a_x)
+ \frac{1}{2}(\min\{x_k - \lambda_x, x_j - \lambda_x, \lambda_xa\})^2 - (x_j - \lambda_xa)^2 +
(x_j - a_x)(\lambda_x - a_x)(\min\{x_k - \lambda_x, a_x + \lambda_x\} - x_j + \lambda_x)
+ \frac{1}{2}(\min\{x_k - \lambda_x, a_x + \lambda_x\})^2 - (x_j - \lambda_x)^2 +
(x_j - a_x)(\lambda_x - a_x)(\min\{x_k - \lambda_x, a_x + \lambda_x\} - x_j + \lambda_x)
+ \frac{1}{2}(\min\{x_k - \lambda_x, a_x + \lambda_x\} - x_j + \lambda_x)

\text{N-D case}

\text{QQ}_k(x_j, x_k) = \prod_{i=1}^{N} a_i^{-1} \left\{ (x_j - a_i)(x_k - a_i)(a_i + 2\lambda_i - \max\{x_k - a_i, x_j - a_i\}) \right\}
+ \frac{1}{2}(a_i + 2\lambda_i - \max\{x_k - a_i, x_j - a_i\})
+ \frac{1}{2}(\min\{x_k - a_i, x_j - a_i\} + \lambda_i)^2 - \max\{x_k - a_i, x_j - a_i - \lambda_i, a_i + \lambda_i\}^2 -
\frac{1}{2}(x_j - a_i)(\min\{x_k - a_i, x_j - a_i + \lambda_i\})^2 - \max\{x_k - a_i, x_j - a_i - \lambda_i, a_i + \lambda_i\}^2
+ (\lambda_i^2 + x_k\lambda_i + x_j\lambda_i + x_jx_k)(\min\{x_k + \lambda_i, x_j + \lambda_i\} - \max\{x_k - \lambda_i, x_j - \lambda_i - \lambda_i, a_i + \lambda_i\}^3 -
\frac{1}{2}(x_j + x_k + 2\lambda_i)(\min\{x_k + \lambda_i, x_j + \lambda_i\})^2 - \max\{x_k + \lambda_i, x_j - \lambda_i, a_i + \lambda_i\}^2
+ (\lambda_i^2 + x_k\lambda_i + x_j\lambda_i + x_jx_k)(\min\{x_k + \lambda_i, x_j + \lambda_i\} - \max\{x_k - \lambda_i, x_j - \lambda_i - \lambda_i, a_i + \lambda_i\}^3 -
\frac{1}{2}(x_j - a_i)(\min\{x_k - a_i, a_i + \lambda_i\})^2 - (x_j - \lambda_i)^2 +
(x_j - a_i)(\lambda_i - a_i)(\min\{x_k - a_i, a_i + \lambda_i\} - x_j + \lambda_i)
+ \frac{1}{2}(\min\{x_k - a_i, a_i + \lambda_i\} - x_j + \lambda_i)
\[ + 2\lambda(\{x_{j,(i)} + \lambda_{(i)}(\min\{x_{k,(i)} - \lambda_{(i)} - x_{j,(i)} + \lambda_{(i)}\}) - \max\{x_{j,(i)} - \lambda_{(i)} - a_{(i)} + \lambda_{(i)}\}) - \frac{1}{2}(\min\{x_{k,(i)} - \lambda_{(i)} - x_{j,(i)} + \lambda_{(i)}\})^2 - \max\{x_{j,(i)} - \lambda_{(i)} - a_{(i)} + \lambda_{(i)}\}^2) \\
1_{0, \infty}(\min\{x_{k,(i)} - \lambda_{(i)} - x_{j,(i)} + \lambda_{(i)}\}) - \max\{x_{j,(i)} - \lambda_{(i)} - a_{(i)} + \lambda_{(i)}\} \} \]

\[ + (x_{k,(i)} - a_{(i)})(\frac{1}{2}(\min\{x_{j,(i)} - \lambda_{(i)} - a_{(i)} + \lambda_{(i)}\})^2 - (x_{k,(i)} - \lambda_{(i)}))^2 + (\lambda_{(i)} - a_{(i)})(\min\{x_{j,(i)} - \lambda_{(i)} - a_{(i)} + \lambda_{(i)}\} - x_{k,(i)} + \lambda_{(i)}) \\
1_{0, \infty}(\min\{x_{k,(i)} + \lambda_{(i)} - x_{j,(i)} + \lambda_{(i)}\} - \max\{x_{k,(i)} + \lambda_{(i)} - a_{(i)} + \lambda_{(i)}\}) \\
1_{0, \infty}(\frac{1}{2}(\min\{x_{k,(i)} + \lambda_{(i)} - x_{j,(i)} + \lambda_{(i)}\})^2 - (x_{k,(i)} + \lambda_{(i)}))^2 + (\lambda_{(i)} + a_{(i)})(\min\{x_{j,(i)} + \lambda_{(i)} - x_{j,(i)} - \lambda_{(i)}\} - \max\{x_{k,(i)} + \lambda_{(i)} - a_{(i)} + \lambda_{(i)}\}) \\
1_{0, \infty}(\min\{x_{k,(i)} + \lambda_{(i)} - x_{j,(i)} + \lambda_{(i)}\} - \max\{x_{k,(i)} + \lambda_{(i)} - a_{(i)} + \lambda_{(i)}\}) \]

\[ + \frac{1}{3}(\min\{x_{k,(i)} - \lambda_{(i)} - x_{j,(i)} - \lambda_{(i)} + a_{(i)} + \lambda_{(i)}\}^3 - (a_{(i)} - \lambda_{(i)}a)^3) + (\lambda_{(i)} + a_{(i)})(\min\{x_{k,(i)} - \lambda_{(i)} - x_{j,(i)} - \lambda_{(i)} + a_{(i)} + \lambda_{(i)}\}^2 - (a_{(i)} - \lambda_{(i)}a)^2) + (\lambda_{(i)} + a_{(i)})(\min\{x_{k,(i)} - \lambda_{(i)} - x_{j,(i)} - \lambda_{(i)} + a_{(i)} + \lambda_{(i)}\} - a_{(i)} + \lambda_{(i)}) \\
1_{0, \infty}(\min\{x_{k,(i)} - \lambda_{(i)} - x_{j,(i)} - \lambda_{(i)} + a_{(i)} + \lambda_{(i)}\} - a_{(i)} + \lambda_{(i)}) \]

\[ + 4\lambda_{(i)}^2(\min\{x_{k,(i)} - \lambda_{(i)} - x_{j,(i)} - \lambda_{(i)}\} - \lambda_{(i)} - a_{(i)}) \\
1_{0, \infty}(\min\{x_{k,(i)} - \lambda_{(i)} - x_{j,(i)} - \lambda_{(i)}\} - \lambda_{(i)} - a_{(i)}) \]  

B.5.2 Indefinite Integral

1-D case

\[ \int_{R}^{Q_{\lambda}(t_{j}, t_{k})} = \alpha^{-1} \]
\[ \int_{-\infty}^{\infty} Q_{\lambda}(t_{j}, t_{k}) Q_{\lambda}(t_{k}, z) dz = \alpha^{-1} \]
\[ \int_{-\infty}^{\infty} 4\lambda^2 dz = 4\lambda^2 \]