Role of Majorization in “Learning the Kernel” within a Gaussian Process Regression Framework

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Prasenjit Kapat, B.Stat., M.Stat. (India)

Graduate Program in Statistics

The Ohio State University

2011

Dissertation Committee:

Prof. Prem K. Goel, Advisor

Prof. Tao Shi

Prof. Radu Herbei
Abstract

Over the recent years, machine learning techniques have breathed a new life into the classical regression framework. The primary focus in these techniques has often been the predictive performance of the estimated models and the models themselves have developed into sophisticated non-linear predictive machines. In this development, the ubiquitous “kernel-trick” has played a very important role by providing a means to compute the inner products in the unwieldy high-dimensional spaces via simple and easily computable functions on the low-dimensional covariate domains, called as kernels. The domain knowledge of data dictates the collection of kernels suitable for the specific application. In “learning the kernel” paradigm, current state of the art is to use some optimization method to select the best kernel for the data at hand from this collection.

The work in this dissertation assumes the existence of a “true” underlying process, a Gaussian Process, (defined by a fully specified covariance kernel) for the given data. The Gaussian Process itself is considered as a prior on the reproducing kernel Hilbert space of functions characterized by the associated kernel. The goal is to make suggestions towards developing some diagnostic tools which can be used to hasten the kernel learning process. In particular, the setup for computational experimentation

\footnote{such a collection is either an ad hoc class of finitely many kernels or a convex collection of kernels which may include one or more parametric family of kernels, for example, Gaussian kernels with bandwidth as the parameter}
is restricted to a Gaussian Process Regression framework with some “mild station-
arity”\textsuperscript{2} and “closure”\textsuperscript{3} type of assumptions on the possible family of kernels. Tools
are developed based on the generalized cross validation and the functional norm of
the estimated functions. The sign-change behaviors of these tools are exploited for
diagnostic purposes. For the tool based on generalized cross validation, a result is
conjectured based on computational evidence, and partially proved, which attempts
to justify the observed sign-change patterns. Complete proofs for the said result are
given under some special classes of kernels. These sign-change behaviors are intended
to be a “guiding stick” for reducing the computational effort and search space for
“learning the kernel.”

\textsuperscript{2}the kernel value of a point with itself does not depend on the point
\textsuperscript{3}Gram matrices closed under the Schür product
To my dearest mother,

Mrs. Tapati Kapat,

my father,

Mr. Sibapada Kapat,

and my grandfather,

Late. Mr. Naryandas Ghose.
Acknowledgments

First and foremost I would like to express my earnest gratitude towards Prof. Prem Goel for his continuous support and endless patience. Be the late evenings in his office, the unending meetings on the weekends, or the occasional sessions at his home, his availability has been invaluable to me. Most importantly, his eternal optimism in this problem and perseverance towards it has been, at times, my only source of encouragement, and has kept me going. This dissertation would never have seen the light of day without his help and for it, I cannot be thankful enough. I am also grateful to Prof. R. B. Bapat at the Indian Statistical Institute, Delhi, for guiding us towards the tool of majorization among Hadamard product of matrices which was crucial for this research. Along with Prof. Goel, I am also indebted to Prof. Mark McCord from the Civil Engineering Department at OSU for supporting me over the past few years through various GRA positions. I am thankful to my colleagues at J.P.Morgan Chase for giving me the flexibility over the past few months to work on my thesis. I would also like to give a special shout-out to Brian Smith from the computer support team at the Statistics Department. He has always gone out of his way to help me with my idiosyncratic needs. And I am certainly grateful to the Statistics Department for giving me the opportunity to pursue a doctoral degree.

This Ph.D. has been an arduous journey. Yet, I owe a small but wonderful group of friends to make it immensely memorable. I have been rather lucky to have the “quality
Two of them, Niranjan Balachandran and Indrajit Bhattacharya, deserve a special mention. Niranjan, from my very first day here, has had an indelible influence in shaping my outlook and personality: from dinners, movies, and sitcoms to mathematics, music, and philosophy; he has impressed upon me in every thinkable way. Indrajit has been more than a friend, he has been a shadow to me in these last few years. His unpretentious opinions and objective deliberations has helped me in making better decisions. I’ll always cherish the countless hours spent with him at various coffee shops. I am also fortunate to have had the opportunity to share many laughs and unforgettable few trips with Arun Kumar, Mallik Rettiganti, and Aritra Sengupta. Nathan, has been the perfect office mate and has made the days and nights spent writing the thesis in the office more bearable. Among friends from my alma mater, Indian Statistical Institute, I particularly thank Suvajit Samanta for the many encouraging and supportive phone conversations; Sasikiran Goteti and Pritam Ranjan for the wonderful memories; they all made the first few years of my graduate school feel much more homely. Last but not the least, I would like to acknowledge Deepayan Sarkar for introducing and getting me hooked on to Debian and R for life. He has painstakingly guided me through the many hurdles in the initial stages.

Finally, for my parents, Maa and Bapi, I truly cannot articulate my gratitude towards them in a few terse sentences. Not being able to be beside them through all these years and my growingly sparse habit of calling them has been considerably difficult on them. Yet, their understanding and acceptance of my whims and fancies has given me tremendous freedom and strength. What I am today is a tribute to their years of hardships; I can never imagine to repay that. I would also like to thank
my kith and kins at *Amulyadham* for being ever so considerate. Their encouragement and affection has always been truly comforting.
Vita

March 30, 1981 .......................... Born – Howrah, India

1998 – 2001 ............................ B.Stat (Hons), Indian Statistical Institute, Kolkata, India

2001 – 2003 ............................ M.Stat, Indian Statistical Institute, Kolkata, India

2003 – 2011 ............................ Graduate Student, The Ohio State University

Publications


Fields of Study

Major Field: Statistics
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>viii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiii</td>
</tr>
<tr>
<td>1. RKHS Theory, examples in Statistical Learning, Gaussian Processes, Hadamard Products, and Majorization</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Brief Introduction to the RKHS Theory</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Example from Statistical Learning</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Commonly used Kernels in Statistical Learning</td>
<td>13</td>
</tr>
<tr>
<td>1.4 Kernel Smoothing Methods for function estimation in comparison to RKHS methods</td>
<td>16</td>
</tr>
<tr>
<td>1.5 Gaussian Process Regression</td>
<td>19</td>
</tr>
<tr>
<td>1.6 Learning the Kernel and Model Selection</td>
<td>22</td>
</tr>
<tr>
<td>1.6.1 “Learning the kernel”</td>
<td>22</td>
</tr>
<tr>
<td>1.6.2 Model selection techniques</td>
<td>28</td>
</tr>
<tr>
<td>2. Gaussian Process Regression Diagnostics</td>
<td>32</td>
</tr>
<tr>
<td>2.1 GPR and Classical Regularization: a Review</td>
<td>34</td>
</tr>
<tr>
<td>2.2 Preliminary Experiments</td>
<td>37</td>
</tr>
<tr>
<td>2.2.1 Expressions for GCV, Functional Norm, Optimal Cost, and Out-of-sample Predictions</td>
<td>38</td>
</tr>
<tr>
<td>2.2.2 Setup for simulation experiments</td>
<td>40</td>
</tr>
</tbody>
</table>
A.1 GCV and Functional norm on $h \times \lambda$ grid using multiple replications of the measurements .............................................. 100

A.2 Optimal cost and Validation set prediction on $h \times \lambda$ grid using $Y^{[v]}_{i}$ ................................. 110

A.3 GCV and Functional norm at $\lambda = \lambda_0$ using multiple replications of the measurements .............................................. 111

B. Supporting Figures for Chapter 2 (Part II) ................................. 122

B.1 Comparing GCV and Functional norm with their respective expectations under the model space at $\lambda = \lambda_0$ using multiple replications of the measurements .............................................. 122

C. Supporting Figures for Chapter 2 (Part III) ................................. 132

C.1 Visualizing Diagnostic Tools based on GCV and Functional norm on $h \times \lambda$ grid using replicated measurements .............................................. 132

C.2 Visualizing Diagnostic Tools based on GCV and Functional norm at $\lambda = \lambda_0$ using replicated measurements .............................................. 142

Bibliography ................................................................. 152
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 GCV (2.2.1): GCV(h,λ) on a $h \times \lambda$ grid using $Y_{tr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>2.2 Functional norm (2.2.2): $\log \nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{tr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>2.3 GCV (2.2.1) at $\lambda = \lambda_0$: GCV(h, $\lambda_0$) as a function of $\log(h)$ using $Y_{tr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>2.4 Functional norm (2.2.2) at $\lambda = \lambda_0$: $\log \nu(h,\lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$ (vertical) and $\log(n) \approx 5.3$ (horizontal).</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>2.5 Comparing GCV(h, $\lambda_0$) (solid blue curve) and $E_{h,\lambda_0} \text{GCV}(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>2.6 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0} \nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>2.7 Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{tr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.</td>
<td>53</td>
<td></td>
</tr>
</tbody>
</table>
2.8 Diagnostic tool based on functional norm (2.4.2): \( Y(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_{tr}^1 \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary. . . . . . . . . . . . . . 54

2.9 Diagnostic tool based on GCV (2.4.1): \( \Gamma(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{tr}^1 \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . . . . . . . 55

2.10 Diagnostic tool based on functional norm (2.4.2): \( Y(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{tr}^1 \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . . . . . . . 55

2.11 Variability in the diagnostic tool based on GCV, \( \Gamma(h, \lambda_0) \), as a function of \( \log(h) \). Variability is depicted using 1000 repetitions, \( Y_{tr}^1, ... , Y_{tr}^{1000} \). First column: \( 1.5 \leq h \leq 60 \); second column: \( h_0/2 \leq h \leq 2h_0 \). Rows: \( h_0 = 3 \) (top), \( h_0 = 14.6 \) (middle), and \( h_0 = 26.1 \) (bottom). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . . . . . . . 58

2.12 Variability in the diagnostic tool based on functional norm, \( Y(h, \lambda_0) \), as a function of \( \log(h) \). Variability is depicted using 1000 repetitions, \( Y_{tr}^1, ... , Y_{tr}^{1000} \). First column: \( 1.5 \leq h \leq 60 \); second column: \( h_0/2 \leq h \leq 2h_0 \). Rows: \( h_0 = 3 \) (top), \( h_0 = 14.6 \) (middle), and \( h_0 = 26.1 \) (bottom). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . . . . . . . 59

3.1 Expected diagnostic tool based on GCV (3.3.1): \( E_{h_0,\lambda_0} \Gamma(h, \lambda_0) \) as a function of \( \log(h) \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . . . . . . . 75

3.2 Expected diagnostic tool based on functional norm (3.3.2): \( E_{h_0,\lambda_0} Y(h, \lambda_0) \) as a function of \( \log(h) \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . . . . . . . 76

3.3 Expected diagnostic tool based on GCV (3.3.1): \( E_{h_0,\lambda_0} \Gamma(h, \lambda) \) on \( h \times \lambda \) grid. Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). . . . . . . . . . . . . . . . . 78

3.4 Expected diagnostic tool based on functional norm (3.3.2): \( E_{h_0,\lambda_0} Y(h, \lambda) \) on \( h \times \lambda \) grid. Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). . . . . . . . . . . . . . . . . 78
3.5 Expected diagnostic tool based on GCV (3.3.1): $E_{h_0, \lambda_0} T(h, \lambda_0)$ as a function of $\log(h/h_0)$. Here $h = 33.69$ was kept fixed. The subplots in each row correspond to different noise levels, specified at the top. There are 10 different curves in each subplot corresponding to independent sets of training points. The bottom row zooms in on the top row such that $-1 \leq \log(h/h_0) \leq 1$.

3.6 Expected diagnostic tool based on functional norm (3.3.2): $E_{h_0, \lambda_0} T(h, \lambda_0)$ as a function of $\log(h/h_0)$. Here $h = 1$ was kept fixed. The subplots in each row correspond to different noise levels, specified at the top. There are 10 different curves in each subplot corresponding to independent sets of training points. The bottom row zooms in on the top row such that $-1 \leq \log(h/h_0) \leq 1$.

A.1 GCV (2.2.1): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y_{tr}^2$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).

A.2 GCV (2.2.1): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y_{tr}^3$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).

A.3 GCV (2.2.1): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y_{tr}^3$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).

A.4 GCV (2.2.1): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y_{tr}^5$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).

A.5 GCV (2.2.1): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y_{tr}^6$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).

A.6 GCV (2.2.1): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y_{tr}^7$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).

A.7 GCV (2.2.1): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y_{tr}^8$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).
| A.8 GCV (2.2.1): GCV($h,\lambda$) on a $h \times \lambda$ grid using $Y_{tr}^{16}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 104 |
| A.9 GCV (2.2.1): GCV($h,\lambda$) on a $h \times \lambda$ grid using $Y_{tr}^{10}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 104 |
| A.10 Functional norm (2.2.2): log $\nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{tr}^{2}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 105 |
| A.11 Functional norm (2.2.2): log $\nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{tr}^{3}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 106 |
| A.12 Functional norm (2.2.2): log $\nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{tr}^{4}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 106 |
| A.13 Functional norm (2.2.2): log $\nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{tr}^{5}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 107 |
| A.14 Functional norm (2.2.2): log $\nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{tr}^{6}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 107 |
| A.15 Functional norm (2.2.2): log $\nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{tr}^{7}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 108 |
| A.16 Functional norm (2.2.2): log $\nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{tr}^{8}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 108 |
| A.17 Functional norm (2.2.2): log $\nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{tr}^{9}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). | 109 |
A.18 Functional norm (2.2.2): $\log \nu(h, \lambda)$ on a $h \times \lambda$ grid using $Y_{10}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). ........................................ 109

A.19 Optimal cost (2.2.3): $C(h, \lambda)$ on a $h \times \lambda$ grid using $Y_{10}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). ........................................ 110

A.20 Validation set predictions (2.2.5): MSE$_v(h, \lambda)$ on a $h \times \lambda$ grid using $Y_{10}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). ........................................ 111

A.21 GCV (2.2.1) at $\lambda = \lambda_0$: GCV$_{10}(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{20}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. ........................................ 112

A.22 GCV (2.2.1) at $\lambda = \lambda_0$: GCV$_{10}(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{30}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. ........................................ 112

A.23 GCV (2.2.1) at $\lambda = \lambda_0$: GCV$_{10}(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{40}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. ........................................ 113

A.24 GCV (2.2.1) at $\lambda = \lambda_0$: GCV$_{10}(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{50}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. ........................................ 113

A.25 GCV (2.2.1) at $\lambda = \lambda_0$: GCV$_{10}(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{60}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. ........................................ 114

A.26 GCV (2.2.1) at $\lambda = \lambda_0$: GCV$_{10}(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{70}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. ........................................ 114

A.27 GCV (2.2.1) at $\lambda = \lambda_0$: GCV$_{10}(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{80}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. ........................................ 115

xvii
A.28 GCV (2.2.1) at \( \lambda = \lambda_0 \): GCV \((h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_9^{tr} \).
Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \). ................................. 115

A.29 GCV (2.2.1) at \( \lambda = \lambda_0 \): GCV \((h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_{10}^{tr} \).
Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \). ................................. 116

A.30 Validation set predictions (2.2.5) at \( \lambda = \lambda_0 \): MSE \(_v\) \((h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_1^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines are at \( \log(h_0) \). ................................. 116

A.31 Functional norm (2.2.2) at \( \lambda = \lambda_0 \): \( \log \nu (h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_2^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \) (vertical) and \( \log(n) \approx 5.3 \) (horizontal). ................................. 117

A.32 Functional norm (2.2.2) at \( \lambda = \lambda_0 \): \( \log \nu (h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_3^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \) (vertical) and \( \log(n) \approx 5.3 \) (horizontal). ................................. 117

A.33 Functional norm (2.2.2) at \( \lambda = \lambda_0 \): \( \log \nu (h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_4^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \) (vertical) and \( \log(n) \approx 5.3 \) (horizontal). ................................. 118

A.34 Functional norm (2.2.2) at \( \lambda = \lambda_0 \): \( \log \nu (h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_5^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \) (vertical) and \( \log(n) \approx 5.3 \) (horizontal). ................................. 118

A.35 Functional norm (2.2.2) at \( \lambda = \lambda_0 \): \( \log \nu (h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_6^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \) (vertical) and \( \log(n) \approx 5.3 \) (horizontal). ................................. 119

A.36 Functional norm (2.2.2) at \( \lambda = \lambda_0 \): \( \log \nu (h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_7^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \) (vertical) and \( \log(n) \approx 5.3 \) (horizontal). ................................. 119

A.37 Functional norm (2.2.2) at \( \lambda = \lambda_0 \): \( \log \nu (h, \lambda_0)\) as a function of \( \log(h) \) using \( Y_8^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \) (vertical) and \( \log(n) \approx 5.3 \) (horizontal). ................................. 120
A.38 Functional norm (2.2.2) at $\lambda = \lambda_0$: $\log \nu(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^9$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$ (vertical) and $\log(n) \approx 5.3$ (horizontal). . . . 120

A.39 Functional norm (2.2.2) at $\lambda = \lambda_0$: $\log \nu(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^9$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$ (vertical) and $\log(n) \approx 5.3$ (horizontal). . . . 121

B.1 Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^2$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. . . . . . . . . 122

B.2 Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^3$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. . . . . . . . . 123

B.3 Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^4$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. . . . . . . . . 123

B.4 Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^5$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. . . . . . . . . 124

B.5 Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^6$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. . . . . . . . . 124

B.6 Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^7$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. . . . . . . . . 125

B.7 Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^8$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. . . . . . . . . 125

B.8 Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^9$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. . . . . . . . . 126
B.9 Comparing GCV($h, \lambda$) (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_{10}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 126

B.10 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_2^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 127

B.11 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_3^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 127

B.12 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_4^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 128

B.13 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_5^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 128

B.14 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_6^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 129

B.15 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_7^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 129

B.16 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_8^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 130

B.17 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_9^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 130

B.18 Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y_{10}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$). . . . . . . . . 131
C.1 Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{3r}$.  
Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary. . . . . . . . . . . . . . . . . . . . . 132

C.2 Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{3r}$.  
Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary. . . . . . . . . . . . . . . . . . . . . 133

C.3 Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{3r}$.  
Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary. . . . . . . . . . . . . . . . . . . . . 133

C.4 Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{3r}$.  
Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary. . . . . . . . . . . . . . . . . . . . . 134

C.5 Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{3r}$.  
Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary. . . . . . . . . . . . . . . . . . . . . 134

C.6 Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{3r}$.  
Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary. . . . . . . . . . . . . . . . . . . . . 135

C.7 Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{3r}$.  
Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary. . . . . . . . . . . . . . . . . . . . . 135

C.8 Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{3r}$.  
Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary. . . . . . . . . . . . . . . . . . . . . 136
| C.9 | Diagnostic tool based on GCV [2.4.1]: \( \Gamma(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_{10}^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary. |
| C.10 | Diagnostic tool based on functional norm [2.4.2]: \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_{2}^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary. |
| C.11 | Diagnostic tool based on functional norm [2.4.2]: \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_{3}^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary. |
| C.12 | Diagnostic tool based on functional norm [2.4.2]: \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_{4}^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary. |
| C.13 | Diagnostic tool based on functional norm [2.4.2]: \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_{5}^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary. |
| C.14 | Diagnostic tool based on functional norm [2.4.2]: \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_{6}^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary. |
| C.15 | Diagnostic tool based on functional norm [2.4.2]: \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_{7}^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary. |
| C.16 | Diagnostic tool based on functional norm [2.4.2]: \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_{8}^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary. |
C.17 Diagnostic tool based on functional norm (2.4.2): $\mathcal{Y}(h, \lambda)$ on $h \times \lambda$ grid using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.

C.18 Diagnostic tool based on functional norm (2.4.2): $\mathcal{Y}(h, \lambda)$ on $h \times \lambda$ grid using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.

C.19 Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

C.20 Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

C.21 Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

C.22 Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

C.23 Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

C.24 Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

C.25 Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

C.26 Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
C.27 Diagnostic tool based on GCV (2.4.1): \( \hat{\Gamma}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 146

C.28 Diagnostic tool based on functional norm (2.4.2): \( \check{\Upsilon}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{3tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 147

C.29 Diagnostic tool based on functional norm (2.4.2): \( \check{\Upsilon}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{4tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 148

C.30 Diagnostic tool based on functional norm (2.4.2): \( \check{\Upsilon}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{5tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 148

C.31 Diagnostic tool based on functional norm (2.4.2): \( \check{\Upsilon}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{6tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 149

C.32 Diagnostic tool based on functional norm (2.4.2): \( \check{\Upsilon}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{7tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 149

C.33 Diagnostic tool based on functional norm (2.4.2): \( \check{\Upsilon}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{8tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 150

C.34 Diagnostic tool based on functional norm (2.4.2): \( \check{\Upsilon}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{9tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 150

C.35 Diagnostic tool based on functional norm (2.4.2): \( \check{\Upsilon}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{10tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 151

C.36 Diagnostic tool based on functional norm (2.4.2): \( \check{\Upsilon}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_{11tr}^c \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal). . . . . . . . . . . 151
Chapter 1: RKHS Theory, examples in Statistical Learning, Gaussian Processes, Hadamard Products, and Majorization

Over the past couple of decades, predictive learning algorithms have been the focus of a lot of research both in the field of Statistics ("statistical learning") and Computer Science ("data mining"). The remarkable success of these algorithms were fuelled by the growing computational capabilities and the mathematical theory of Reproducing Kernel Hilbert Spaces (RKHS) put forth by Aronszajn (1943, 1950) and introduced to the Statistical community by Kimeldorf and Wahba (1971).

The current chapter describes the underlying setup for "learning the kernel" problem and states some of the necessary results from literature that are needed for the current work. It is structured into a few sections: introducing the RKHS theory, giving some examples from the statistical literature where this framework is used, and the various types of kernels used in practice. It also briefly describes the setup for Gaussian Processes, the framework used in this research, and its connection to reproducing kernel Hilbert spaces. Following this, a little discussion on the kernel estimation literature in general and finally some results on Hadamard product of matrices and majorization of vectors is introduced.
The next chapter discusses this research in detail with a Proposition on generalized cross validation for diagnostics followed by the discussion from simulation experiments.

Some concluding remarks and a little discussion on the future avenues of research in the final chapter.

1.1 Brief Introduction to the RKHS Theory

To begin with, a quick introduction to the composition of a generic Hilbert space follows.

A metric space is a non-empty set equipped with a concept of distance ("metric") which is suitable for the treatment of convergent sequences and continuous functions defined on the set. A metric, \( d(\cdot, \cdot) \), is a real-valued bivariate function on this set which is non-negative, symmetric, and satisfies the triangle inequality. Utilizing this metric, Cauchy sequences and their completeness (all Cauchy sequences converge) properties are defined. Additionally, this metric induces a topology of open sets which is then used to introduce separability. (Separability provides a countable cover for the metric space.) Note that, none of the above concepts need any algebraic structure on the metric space.

Assume that the non-empty set is in fact a vector space (linear space). This provides the framework to define norms, \( \| \cdot \| \). A norm is a univariate real-valued function which, similar to metric, is non-negative, scalar multiplicative, and satisfies the triangle inequality. Equipped with a norm, this space is called a normed linear space. Such a norm function, in turn induces a metric, \( d(\cdot, \cdot) = \| \cdot - \cdot \| \). If the
resulting metric space is complete, it is called a *Banach space* (a complete normed linear space).

The main geometric concept missing from the above mathematical structures is that of angles between two vectors, which translates to the concept of *inner products*. An inner product, $\langle \cdot, \cdot \rangle$, is a symmetric bivariate bi-linear function on a normed linear space. The inner product of a point with itself can be defined as the square norm of that point. In this respect, a real (or complex) Banach space whose norm arises from an inner product is termed as a *Hilbert space*.

Now, let $\mathcal{X}$ be a non-empty abstract set (the domain). Let $\mathcal{H}$ be a vector space of real-valued functions defined on $\mathcal{X}$. Assume that $\mathcal{H}$ is endowed with a Hilbert space structure where the inner product is denoted by $\langle \cdot, \cdot \rangle_\mathcal{H}$ and the associated norm, denoted by $\| \cdot \|_\mathcal{H}$, is defined as $\|f\|_\mathcal{H} = (f, f)_\mathcal{H}^{1/2}$. As introduced below, RKH spaces are a special type of Hilbert spaces which can be characterized by “kernel” functions.

**Definition 1** (Reproducing Kernel). A symmetric bivariate function

$$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

$$(x, y) \mapsto \kappa(x, y)$$

is a reproducing kernel of the Hilbert space $\mathcal{H}$ if and only if

(a) $\kappa(\cdot, x) \in \mathcal{H}, \forall x \in \mathcal{X}$, and

(b) $\langle f(\cdot), \kappa(\cdot, x) \rangle_\mathcal{H} = f(x), \forall x \in \mathcal{X}, f \in \mathcal{H}$.

The last condition basically states that the value of the function $f$ at a point $x$ can be computed from the inner product of $f$ with $\kappa(\cdot, x)$, hence the reproducing property. Then, by definition,

$$\kappa(x, y) = \langle \kappa(\cdot, x), \kappa(\cdot, y) \rangle_\mathcal{H}, \quad \forall x, y \in \mathcal{X}.$$
**Definition 2** (Reproducing Kernel Hilbert Space (RKHS)). A Hilbert space, $\mathcal{H}$, of real-valued functions spanned by a reproducing kernel, $\kappa$, is called a reproducing kernel Hilbert Space (RKHS) and is denoted by $\mathcal{H}_\kappa$.

Note that the above definitions do not provide any uniqueness between a reproducing kernel, $\kappa$, and the RKHS $\mathcal{H}_\kappa$. The following theorems build towards that goal. A slightly alternative construction of the theory of kernels and reproducing kernel Hilbert spaces is given in Steinwart et al. (2006, Sections I and II).

A cornerstone in the development of RKHS is the Riesz representation theorem which essentially states the following

**Theorem 1** (Riesz representation). For every continuous functional $L$ on a Hilbert space $\mathcal{H}$, there exists a unique $g_L \in \mathcal{H}$ such that $L(f) = \langle g_L, f \rangle$, $\forall f \in \mathcal{H}$. (See, Gu (2002, Thm. 2.2).)

**Theorem 2.** A Hilbert space of real valued functions on $\mathcal{X}$ has a reproducing kernel if and only if for all $x \in \mathcal{X}$, the Dirac functionals $\delta_x : \mathcal{H} \to \mathbb{R}$ defined by $\delta_x(f) := f(x)$, $f \in \mathcal{H}$, is continuous on $\mathcal{H}$.

A proof of the above theorem is given in Berlinet and Thomas-Agnan (2004, Thm. 1, p. 9) or Steinwart et al. (2006, Defn. 2). An immediate consequence of the above theorem, is the following remarkable and important property of RKHS.

**Corollary 1.** In a RKHS a sequence converging in the norm sense converges pointwise to the same limit.

This is easy to verify; for a reference, see Berlinet and Thomas-Agnan (2004, Cor. 1, p. 10) or Steinwart et al. (2006, Eqn. (2)). For the RKHS theory of complex
valued functions, the kernel has to be of positive type. The analogous concept in the current context of real-valued functions is positive definiteness.

**Result 1.** Any reproducing kernel corresponding to a RKHS is a positive definite function.

The following theorem provides the uniqueness between $\kappa$ and $H_\kappa$. It also gives a way to construct $H_\kappa$ from the linear spans of $\kappa(\cdot, \cdot)$.

**Theorem 3.** [Moore-Aronszajn Theorem]

(a) Let $\kappa$ be a symmetric positive definite function on $\mathcal{X} \times \mathcal{X}$. There exists only one Hilbert space $H_\kappa$ of functions on $\mathcal{X}$ with $\kappa$ as the reproducing kernel.

(b) Let $H_0$ be the linear span of the functions $\{\kappa(\cdot, x) : x \in \mathcal{X}\}$. Then this subspace $H_0$ is dense in $H$, the set of functions on $\mathcal{X}$ which are pointwise limits of Cauchy sequences in $H_0$, with the inner product defined as

$$
\langle \sum_{i=1}^{n} \alpha_i \kappa(\cdot, x_i), \sum_{j=1}^{m} \beta_j \kappa(\cdot, y_j) \rangle_{H_0} = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j \kappa(x_i, y_j)
$$

for all $n, m \in \mathbb{N}$ and for all $\{x_i\}_{1}^{n}, \{y_j\}_{1}^{m} \in \mathcal{X}$.  

For a proof see [Berinet and Thomas-Agnan (2004, Thm. 3, p. 20)] or [Shawe-Taylor and Cristianini (2004, pp. 61–63)] or [Wahba (1990, Thm. 1.1.1)].

The following two theorems give different ways to characterize a positive-definite function. In fact, these characterizations, as stated later, are used as definitions for kernel in a parallel development of the RKHS theory.

**Theorem 4.** A symmetric real-valued function defined on $\mathcal{X} \times \mathcal{X}$ is a reproducing kernel or a positive definite function if and only if there exists a mapping $T$ from $\mathcal{X}$
to some square-summable space $l^2(A)$ such that

$$\kappa(x, y) = \langle T(x), T(y) \rangle_{l^2(A)} = \sum_{\alpha \in A} (T(x))_\alpha (T(y))_\alpha, \quad \forall (x, y) \in \mathcal{X} \times \mathcal{X}. $$

For a proof see [Berlinet and Thomas-Agnan (2004, Thm. 4, p. 22)] which uses the fact that any Hilbert space is isometric to some square-summable space $l^2(A)$ over some set $A$. In fact, this statement is used as a definition for kernel in [Herbrich (2001, Defn. 2.14)]. An analogous statement which can also be used as a definition for kernels (see [Steinwart et al. 2006, Defn. 1]), is the following:

**Theorem 5.** A symmetric real-valued function defined on $\mathcal{X} \times \mathcal{X}$ is positive semi-definite if and only if there exists a Hilbert space, $\mathcal{H}$, of real-valued functions and an associated map (called a feature map), $\phi : \mathcal{X} \rightarrow \mathcal{H}$ such that

$$\kappa(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}. $$

**Proof.** See [Shawe-Taylor and Cristianini (2004, Thm. 3.11)].

The following theorem provides the eigen-expansion of a kernel in a uniformly convergent series and can be theoretically used to construct a feature space for a valid kernel, see [Herbrich (2001, Thm. 2.17)] or [Shawe-Taylor and Cristianini (2004, Thm. 3.13)] or [Berlinet and Thomas-Agnan (2004, Thm. 40, p. 68)].

**Theorem 6.** [Mercer’s theorem] Suppose, $\mathcal{X}$ is subset of an Euclidean space and $\kappa \in L_\infty(\mathcal{X} \times \mathcal{X})$ is a symmetric function, such that the integral operator, $T_\kappa : L^2(\mathcal{X}) \rightarrow L^2(\mathcal{X})$ defined by

$$(T_\kappa(f))(\cdot) = \int_X \kappa(\cdot, x)f(x)dx$$

is positive semi-definite, that is,

$$\int_X \int_X \kappa(x, y)f(x)f(y)dxdy \geq 0, \quad \forall f \in L^2(\mathcal{X}).$$
Let $\psi_i \in L^2(\mathcal{X})$ be the eigenfunction of $T_\kappa$ associated with the eigenvalue $\lambda_i \geq 0$ and normalized such that $\|\psi\| = \int_{\mathcal{X}} \psi^2(x) dx = 1$, that is,

$$
\int_{\mathcal{X}} \kappa(x,y) \psi_i(x) dx = \lambda_i \psi_i(y), \quad \forall y \in \mathcal{X}.
$$

Then,

(a) $(\lambda_i)_{i \in \mathbb{N}} \in l_1$.

(b) $\psi_i \in L_\infty(\mathcal{X})$.

(c)

$$
\kappa(x,y) = \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(y), \quad \forall x, y \in \mathcal{X}.
$$

Using the above decomposition, with $\phi(x) = (\sqrt{\lambda_1} \psi_1(x), \sqrt{\lambda_2} \psi_2(x), \ldots)'$, it is easy to see that $\kappa(x,y) = \langle \phi(x), \phi(y) \rangle_{l_1}$. Thus $\phi$ provides one choice of feature map for the kernel, $\kappa$.

And finally, the term Mercer’s Kernel is used to represent any symmetric, bivariate, positive semi-definite function, $\kappa$, such that every $n \times n$ finite realization matrix of $\kappa$ is positive semi-definite (see Herbrich 2001, Prop. 2.18).

### 1.2 Example from Statistical Learning

To improve the generalization properties of the usual non-linear regression techniques, considerable research has been performed over the last five decades resulting in methods such as the support vector machines, regularization networks, Gaussian processes models, and spline methods. The generic formulation in all these techniques is the following. Consider a training data,

$$
\mathcal{T} := \{(y_i, x_i) : i = 1, \ldots, n\}.
$$
The goal is to minimize the following regularized formulation of regression:

\[
\min_{f \in \mathcal{H}_\kappa} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \Omega(\|f\|_{\mathcal{H}_\kappa}) \quad (1.2.1)
\]

where \(\mathcal{H}_\kappa\) is some RKHS, ideally chosen depending on the particular problem at hand, and \(\Omega(\cdot)\) is a roughness penalty function. The underlying assumption is that the “true” data generating model, in cases where it makes sense, belongs to \(\mathcal{H}_\kappa\). In other cases, \(\mathcal{H}_\kappa\) is understood as the chosen search space for a function which sufficiently explains the data and reliably predicts future observations as well.

In their seminal paper [Kimeldorf and Wahba (1971)], provided a solution, known as the Representer Theorem, for specific choices of the input domain (a closed interval in \(\mathbb{R}\)), of the loss functions (weighted squared error), and of the penalty function (square integrable, \(m\)-order differential operator); it was later generalized in [Wahba (1999)]. Recently [Schölkopf and Smola (2002)] provided a review of subsequent refinements where they (also see [Schölkopf et al. (2001)]) gave a simple proof in a very general framework for any loss function, with the only restriction that the penalty function, \(\Omega(\cdot)\), be real valued and strictly monotonically increasing. The solution to the problem (1.2.1) is given by

**Theorem 7.** [Generalized Representer theorem] (positive definite case) Let the loss function \(L(y_i, f)\) be a functional of \(f\) which depends on \(f\) only pointwise, that is, through \(\{f(x_i)\}_{i=1}^{n}\). Then any solution to (1.2.1) has the form

\[
f(x) = \alpha_0 + \sum_{i=1}^{n} \alpha_i \kappa(x_i, x), \quad \text{where } \alpha_i \in \mathbb{R}, \forall, i.
\]

In the above theorem, \(\kappa\) was assumed to be positive definite, that is, the null space of the corresponding RKHS \(\mathcal{H}_\kappa\) is just the singleton \(\{f \equiv 0\}\). In the general case, with
positive semi-definite, $\kappa$, $\mathcal{H}_\kappa$ will have non-trivial null space $\{f \in \mathcal{H}_\kappa : \|f\|_{\mathcal{H}_\kappa} = 0\}$, and the representer theorem states that the solution is of the form,

$$f(x) = \sum_{i=1}^{n} \alpha_i \kappa(x, x_i) + \sum_{j=1}^{l} \beta_j \eta_j(x),$$

where the functions $\{\eta_j(\cdot)\}_{i=1}^{l}$ span the null space of $\mathcal{H}_\kappa$ and the coefficients, $\alpha_i$ and $\beta_j \in \mathbb{R}$.

Some examples where the above methodology is used in Statistical literature are briefly discussed below.

**Example 1: Support Vector Classifiers.**

Consider the usual support vector machine classifiers, were the responses, $y_i \in \{-1, +1\}$, with the hinge loss, given by

$$L(y_i, f(x_i)) = \max\{0, 1 - y_i f(x_i)\}$$

and the penalty function,

$$\Omega(\|f\|_{\mathcal{H}_\kappa}) = \|f\|_{\mathcal{H}_\kappa}^2.$$ 

If $\Phi(\cdot) = (\phi_1(\cdot), \ldots, \phi_M(\cdot))'$ represents a set of feature vector for $M$ feature maps in $\mathcal{H}_\kappa$ ($M$ can be $\infty$), then the RKHS function can be represented in terms of the kernel as

$$f(x) = \Phi(x)' \beta + \beta_0$$

$$= \sum_{i=1}^{n} \alpha_i y_i \langle \Phi(x), \Phi(x_i) \rangle + \beta_0$$

where $\alpha_i$ are the Lagrange coefficients in the primal formulation. Here $\Omega(\|f\|_{\mathcal{H}_\kappa}) = \|\beta\|^2$, and the estimated classification function is given by,

$$\hat{f}(x) = \sum_{i=1}^{n} \hat{\alpha}_i y_i \kappa(x, x_i) + \hat{\beta}_0.$$
The predicted class is then based on \( \text{sign} \hat{f}(x) \). Three popular choices for \( \kappa \) in the SVM literature are

- **Polynomial:** \( \kappa(x_1, x_2) = (1 + \langle x_1, x_2 \rangle)^p \),

- **RBF (Gaussian Kernel):** \( \kappa(x_1, x_2) = \exp\{-\sigma \|x_1 - x_2\|^2\} \), and

- **Neural networks:** \( \kappa(x_1, x_2) = \tanh(a \langle x_1, x_2 \rangle + b) \).

**Example 2: Smoothing splines.**

Let \( W_m[0, 1] = \{ f : f, f', f'', \ldots, f^{m-1} \text{ absolutely continuous, } f^m \in L_2 \} \) be the \( m \)-th order Sobolev space, \( W^0_m[0, 1] \) be the Hilbert space with an additional boundary condition that the first \( m - 1 \) derivatives vanish at zero. Also let \( \pi_m[0, 1] = \text{span}\{\phi_p(x) = x^p/p!, p = 1, \ldots, m - 1\} \). As noted in [Wahba (1990, p. ??)] all the three spaces, \( W_m[0, 1], W^0_m[0, 1], \) and \( \pi_m[0, 1] \) are RKHS and the reproducing kernel for \( W^0_m[0, 1] \) is given by

\[
\kappa^1(s, t) = \int_0^1 G_m(t, u) G_m(s, u) \, du, \quad \text{for } s, t \in [0, 1],
\]

where \( G_m \) is the Green’s functions,

\[
G_m(t, u) = \frac{(t - u)^{m-1}}{m-1}. 
\]

Then, given the data \( \{(y_i, x_i) : i = 1, \ldots, n\} \), the smoothing splines problem is to find a \( \hat{f}_\lambda(\cdot) \in W_m[0, 1] \) minimizing,

\[
\frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_0^1 (f^m)^2 \, dx.
\]

This can be restated as: find \( \hat{f}_\lambda(\cdot) \in W_m[0, 1] = \pi_m[0, 1] \oplus W^0_m[0, 1] \) minimizing,

\[
\frac{1}{n} \sum_{i=1}^n (y_i - \langle \kappa(x_i, \cdot), f \rangle)^2 + \lambda \|P_1 f\|^2, \quad (1.2.2)
\]
where $P_1$ is the orthogonal projection operator into $W_0^m$ and $\kappa$ is the reproducing kernel of $W_m$. Further, $\kappa$, can be broken down into two parts,

$$\kappa(\cdot, \cdot) = \kappa^0(\cdot, \cdot) + \kappa^1(\cdot, \cdot) =: P_0\kappa + P_1\kappa.$$ 

Then, any $f \in W_m$ can be written as,

$$f(\cdot) = \sum_{p=1}^{m-1} d_p \phi_p(\cdot) + \sum_{i=1}^n c_i \kappa^1(x_i, \cdot) + \rho(\cdot),$$

(1.2.3)

where $\rho$ is some element in $W_m$ orthogonal to $\{\phi_p\}_{1}^{m-1}$ and $\{\kappa^1(x_i, \cdot)\}_{1}^{n}$, and the penalty functional $\|P_1 f\|^2$ can be expressed as,

$$\sum_{i,j=1}^n c_i c_j \kappa^1(x_i, x_j) + \|\rho\|^2.$$  

(1.2.4)

Using (1.2.2), (1.2.3), and (1.2.4), it is clear that for the minimizing $f$, $\|\rho\|^2 = 0$. The minimizer $\hat{f}_\lambda$ is then of the form,

$$\hat{f}_\lambda(\cdot) = \sum_{p=1}^{m-1} \hat{d}_p \phi_p(\cdot) + \sum_{i=1}^n \hat{c}_i \kappa^1(x_i, \cdot).$$

The explicit equations for $\hat{d}_p$ and $\hat{c}_i$ are in terms of the matrices, $[\phi_i(x_j)]$ and $[\kappa^1(x_i, x_j)]$. For reference see Kimeldorf and Wahba (1971) or Wahba (1990, Eqn. 1.3.9).

Example 3: Generalized ridge regression or penalized least squares.

For any chosen kernel function $\kappa$, with squared error loss and square norm as penalty, the infinite dimensional penalized regression problem can be stated as,

$$\min_{\{c_j\}_1} \sum_{i=1}^n \left( y_i - \sum_{j=1}^\infty c_j \phi_j(x_i) \right)^2 + \lambda \sum_{j=1}^\infty \frac{c_j^2}{\gamma_j},$$

where the eigen-expansion of the chosen kernel is assumed to be

$$\kappa(x_1, x_2) = \sum_{j=1}^\infty \gamma_j \phi_j(x_1) \phi_j(x_2).$$

11
The corresponding finite dimensional minimization problem is given by
\[
\min_{\alpha} (y - K\alpha)'(y - K\alpha) + \lambda \alpha'K\alpha
\]
where \( K = [\kappa(x_i, x_j)] \) is the so-called Gram matrix. The solution for \( \alpha \) is given by,
\[
\hat{\alpha} = (K + \lambda I)^{-1} Y.
\]
The fitted values and the prediction function is given by,
\[
\hat{Y}_\lambda = K\hat{\alpha} = (I + \lambda K^{-1})^{-1} Y \quad \text{and} \quad \hat{f}_\lambda(\cdot) = \sum_{i=1}^{n} \hat{\alpha}_i \kappa(x_i, \cdot), \quad (1.2.5)
\]
respectively. The details are given in, e.g., Hastie et al. (2001, Sec. 5.8). The penalty coefficient, \( \lambda \), is generally chosen using some cross-validation techniques and has been discussed considerably in the literature, e.g., Hastie et al. (2001, Sec. 5.5), Gu (2002, Sec. 3.2).

**Example 4:** Bayesian MAP estimators.

Consider the following model for the data and the prior.

- **Likelihood as loss.** Let \( \exp\{-L(y_i, f(x_i))\} \) denote the data model (that is, likelihood), then the loss function part of (1.2.1) is simply the negative log-likelihood of the data.

- **Penalty as prior.** Consider an improper prior on the parameters so that the prior density function is given by, \( \exp\{-\Omega(\|f\|)\} \). Then the penalty is simple the negative log-prior. E.g., Gaussian process priors with covariance function \( \kappa \).

In this case the minimizer, given the data, is the maximum aposteriori (MAP) estimator for the combined data and prior models. This is further discussed in Section 1.5.
1.3 Commonly used Kernels in Statistical Learning

Kernel methods are ubiquitous in many different kind of learning problems. Depending on the type of data these can be classified into regression, classification, and clustering. Although, classification and clustering are closely related. Even otherwise, the methodologies in all most all of the sub contexts where the data is continuous, can be termed as “smoothing.” Genton (2001) provides a nearly exhaustive description of various kernels used in the Statistical Learning literature.

Starting with the linear kernel \( \kappa(x,y) = \langle x, y \rangle \), the polynomial kernels (as discussed in Section 3.1.1) are probably the most simple and comprehensible kernels due to its finite dimensional feature space. Interestingly, for a second order polynomial kernel, \( \langle x, y \rangle^2 \), the feature map given by,

\[
\phi : x \mapsto \phi(x) = (x_i x_j)_{i,j=1}^n \in \mathbb{R}^{n^2},
\]

can be associated with ‘all pairwise products’ and consequently to the idea of \( U \)-statistics. Although such polynomial kernels are enticing, Steinwart (2002) proves that even for simple, noise free classification problems, SVMs with polynomial kernels can have arbitrarily poor performance.

The most prevalent and widely used kernels of all is the Gaussian kernel. Since the feature space is infinite dimensional, it is not as easy to grasp as the polynomials; none-the-less the implementation is as straight forward as any other ‘kernel trick’ method. Especially, in the smoothing literature, the exponential decay of the eigenvalues (in a more general setting see the Gaussian convolution operators as discussed in Shi et al. 2009) and the bell-shaped smoothing effect that it induces over the input domain has made it immensely popular in various applications.
For classification problems, kernel methods have been used where the data is a sequence of texts, such as words for document classification \cite{Joachims1998}, alphabets for genome classification \cite{Watkins2000}, sequence of images for face detection \cite{Romdhani2001} and many many others.

Some kernels are very specifically developed for text classifications \cite[Sec. 10.2]{Shawe-Taylor2004}, such as the vector space kernels,

\[
\kappa(d_1, d_2) = \langle \phi(d_1), \phi(d_2) \rangle = \sum_{i=1}^{n} tf(t_i, d_1) tf(t_i, d_2)
\]

where \( \phi(d) = (tf(t_1, d), \ldots, tf(t_n, d)) \) is the feature map defined using the term-document frequency, \( tf(t, d) \): frequency of term \( t \) in the document \( d \). These kernels are further enhanced by utilizing various weighting schemes such as the inverse document frequency or proximity matrices. Additionally, for specific tasks, these can be non-linearly embedded in higher dimensional feature spaces via other kernels, such as, polynomial,

\[
((\phi(d_1), \phi(d_2)) + 1)^p.
\]

Likewise, for data of strings \cite[Sec. 10.3]{Shawe-Taylor2004}, which are more common in the genetics, e.g., such as DNA sequences, there are specialized kernels, such as, \( p \)-spectrum kernel:

\[
\kappa_p(s, t) = \langle \phi^p(s), \phi^p(t) \rangle = \sum_{u \in \Sigma^p} \phi^p_u(s) \phi^p_u(t),
\]

where \( \Sigma^p \) is the space of strings of length \( p \) and

\[
\phi^p_u(s) = |\{(v_1, v_2) : s = v_1uv_2\}|
\]

gives the number of strings which have a substring of length \( p \) in common. This is generalized to “all subsequences kernel” which includes substrings (as well as subsequences) of arbitrary length in common, and further to “gap-weighted subsequence
kernels,” which weights the occurrences of subsequences according to how spread out they are. Saigo et al. (2004), warns against diagonal dominance of the gram matrices of such string kernels, and suggests using $\log \kappa(\cdot, \cdot)$ even though it is not a Mercer’s kernel anymore.

There is also a broad class of kernels which are defined by probability measures on the input domain (Shawe-Taylor and Cristianini 2004, Sec. 12.1). Some examples include:

(a) $P-$kernels

$$\kappa(x, y) = P(x, y)$$

where $P$ is a “positive semi-definite probability distribution” on $\mathcal{X} \times \mathcal{X}$, 

(b) marginalization (HMM) kernels

$$\kappa(x, y) = \sum_{z \in Z} P(x|z)P(y|z)P_Z(z).$$

This kernel works as in, given $Z, X$ and $Y$ are conditionally independent.

(c) Fisher’s kernels (as described in Section 3.1.1). The most common problem in using the Fisher’s kernel is the calculation of the inverse of the information matrix, so a ‘naive’ approach just ignores this matrix. Even then, since it involves the derivative of the log-likelihood, division by $P(x|\theta)$, can result in “more likely” points having a smaller kernel value than “less likely” points. This is easily overcome by normalizing the kernels (Herbrich 2001, p. 36, Cor. 2.21) or even considering $e^{-x}$ transformation of the kernel.

In the regression context, applications are primarily in the field of non-parametric function estimation, and in that sense it becomes all the more difficult to distinguish it from “kernel smoothing” techniques described in Section 1.4.
The foremost thing to consider, of course, is the type of data, for example, text (email, web-documents, etc.), strings (DNA sequences, peptide sequences, etc.), images (pixels, voxels, etc.), sound (pitch, frequency, etc.), continuous observations (functional responses, time series, etc.). There are kernels suited to (broadly speaking) such specific types of data. Of course, this only reduces the hunt space for kernels, the kernels themselves are still parametric functions and the choice of these parameters itself becomes the interesting problem.

Finally, in all these myriad of methods though, the ultimate goal, very often, is just the prediction – how good does the machine do on new sets? And this leads to regularized-“black-box” solutions. From this perspective, the ubiquitous technique of cross-validation along with regularization, is probably unbeatable; and as argued in [Wahba (1990), Section 4.8] CV procedures should be more robust against model mis-specification.

The model parameters are learned by optimization and the kernel and regularization parameter by cross-validation.

1.4 Kernel Smoothing Methods for function estimation in comparison to RHKS methods

[Parzen (1962)] introduced the concept of kernel density estimation (or Parzen window method) in estimating the density function, $f$, over an input space, given a sample from that space, by

$$
\hat{f}_n(x) = \frac{1}{nh} \sum_{i=1}^{n} \kappa\left(\frac{x - x_i}{h}\right),
$$

where $\kappa$ is some kernel with a bandwidth $h$. Some of the examples of kernels that were considered by Parzen are, $0.5I(|x| \leq 1)$, $(1 - |x|)I(|x| \leq 1)$, $(2\pi)^{-0.5}e^{-0.5x^2}$,
$0.5e^{-|x|}$ and $[\pi (1 + x^2)]^{-1}$. The complete list is given in Parzen (1962, Table 1). The subsequent goal was to investigate the statistical properties of such an estimate and the conditions on $\kappa(\cdot)$ that are needed for the purpose. The idea was to use this method in other situations such as the hazard function, $\frac{f(x)}{1-F(x)}$, or or even in $N-$class classification problems, $\frac{\pi_i f_i(x)}{\sum_{i=1}^{N} \pi_i f_i(x)}$. Note that, there were no further restrictions on the functions (other than being a density function) such as restricting to a particular functional class. Hence, in contrast to the RKHS based methods, the kernels used in these settings, need not necessarily be positive semi-definite in the sense of Mercer’s kernels.

Although, the use of Gaussian mixture models for density estimation, is analogous to these methods, the use of ‘Gaussian kernel’ puts it in a RKHS framework.

Another of the early examples in this regard is the Nadaraya-Watson kernel regression proposed by Nadaraya (1964) where the goal is to estimate the conditional expectation of the observation given the predictors,

$$E(Y|X) = f(X);$$

where the function $f$ is estimated as a weighted average of the responses using a kernel $\kappa_h$ of bandwidth $h$,

$$\hat{f}_h(x) = \sum_{i=1}^{n} \frac{\kappa_h(x-x_i))}{\sum_{j=1}^{n} \kappa_h(x-x_j))} Y_i$$

Two other similar estimators of the regression function are the Priestly-Chao kernel estimator and Gasser-Müller kernel estimator; see Simonoff (1996).

A slightly different approach is the weighted least-squares technique where the weights are given by some parametric kernel function (see, for example Hastie et al.)
Examples include local (at $x_0$) polynomial regression,

$$\min \sum_{i=1}^{n} \kappa_h(x_0, x_i) \left( y_i - \alpha(x_0) - \sum_{j=1}^{p} \beta_j(x_0)x_i^j \right)^2,$$

or local multiple regression (replace $x_i^j$ by $x_{ij}$ above). The analogous methodologies in the RKHS setup include the kernel regression (see Example 4 in Section 1.2). Additional concepts of kernel partial least squares and kernel PCA can be seen in Rosipal and Trejo (2002). On the topic of variable selection in the feature space, Blanchard et al. (2007) provides a comprehensive survey of the statistical properties of kernel PCA.

Even the $m$--nearest-neighbor methods can be considered as kernel smoothing, with the following kernel ($x_m$, is the $m$--th closest point to $x$):

$$\kappa(x, y) = I \left( \frac{|x - y|}{|x - x_m|} \leq 1 \right)$$

Again, this is not a positive semi-definite kernel. In fact, this not even a kernel in the usual sense of bivariate function, it kernel value between two points involves the other data points as well. But this can be extended to the usual RKHS setup by computing the nearest neighbors in the feature space (Yu et al. 2002).

Thus, as opposed to the kernels of RKHS, the usual “smoothing-kernels” don’t need to satisfy the positive semi-definite condition. Generally, symmetry about zero, and in cases of density estimation, finite integral over the input domain, are the only major requirements of such smoothing-kernels. The main benefit of the RKHS setup is the linearity in the high dimensional feature space, the inner-products in which can be simply computed using the kernels in the input space. Moreover, the RKHS setup provides a rich mathematical framework for calculation of various error bounds.
on different procedures. See Bousquet et al. (2004) for a comprehensive material on error bounds for classification-centric learning methods.

Some of the other kernels used in the literature are:

- **Triangular**: $\kappa_\theta(x) = (\theta - |x|)I (|x| \leq \theta)$
- **Quartic**: $\kappa_\theta(x) = (\theta^2 - x^2)^2 I (|x| \leq \theta)$
- **Triweight**: $\kappa_\theta(x) = (\theta^2 - x^2)^3 I (|x| \leq \theta)$
- **Cosine**: $\kappa(x) = (\pi/4) \cos(\pi x/2) I (|x| \leq \theta)$
- **Uniform (naive) kernels**: $\kappa_\theta(x) = I(|x| \leq \theta)$ for $\theta \in \mathbb{R}^+$
- **Epanechnikov kernels**: $\kappa_\theta(x) = (\theta - |x|)I(|x| \leq \theta)$ for $\theta \in \mathbb{R}^+$.
- **Rational Quadratic kernels**: $\kappa_\theta(x) = (1 + x^2 \theta_1^{-1} \theta_2^{-2})^{-\theta_1}$ for $\theta = (\theta_1, \theta_2) \in \mathbb{R}^+$
- **Powered Exponential kernels**: $\kappa_\theta(x) = \exp\{-x/\theta_1^{\theta_2}\}$ for $\theta_1 > 0$ and $0 < \theta_2 \leq 2$.

### 1.5 Gaussian Process Regression

This section introduces the Gaussian Process (GP) priors in the context of a regression framework and its association with RKH spaces. Gaussian Processes have long been used by researchers both in the framework of regression and classification (see, e.g., Rasmussen and Williams 2006). For example, in the field of Spatial Statistics, kriging is essentially a form of Gaussian Process regression (GPR) without penalization (Cressie 1993, Sec. 3.4). There is even a vast theory from the time series literature under the nomenclature of Gaussian Markov Models, especially suited
for one-dimensional index sets. In fact, as noted in Rasmussen and Williams (2006, Section 2.8), GP predictions have long been used in time series and geostatistical applications long before it was used in the context of machine learning problems. Kakade et al. (2006) argues that under an online (sequential) prediction setting with log loss, GPR models perform favorably even under rather general conditions such as when no probabilistic assumptions about the data generation process are made. GP models have also been applied in the context of functional data analysis (Shi and Choi 2011).

The basic idea of a GPR model is to consider a nonlinear model as a function in some functional space and assume a GP prior distribution over this space.

Let \( \{ f_x : \Omega \to V, x \in \mathcal{X} \} \), be a stochastic process defined on some abstract probability space \((\Omega, \mathcal{F}, \mathbb{P})\), indexed by an arbitrary set \( \mathcal{X} \), and taking values in \( V \). In the regression framework, generally, \( \mathcal{X} = \mathbb{R}^d \) is taken as the index set and \( V = \mathbb{R} \) is taken as the co-domain of the random variables, \( f_x \).

**Definition 3** (Gaussian Process). The process, \( \{ f_x, x \in \mathcal{X} \} \), is called a Gaussian Process if for any \( n \in \mathbb{N} \) and any \( x_1, \ldots, x_n \in \mathcal{X} \), the probability measure induced by the multivariate random variable, \( (f_{x_1}, \ldots, f_{x_n})' \) on \( \mathcal{V}^n (= \mathbb{R}^n) \) is multivariate Gaussian (Normal) distribution.

In particular, given a mean function \( \mu : \mathbb{R}^d \to \mathbb{R} \), and a covariance function \( \kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \), \( (f_{x_1}, \ldots, f_{x_n})' \sim \text{Gau}_n(\mu, K) \) where \( \mu = (\mu(x_1), \ldots, \mu(x_n))' \) and \( K \) is the Gram matrix \( K_{ij} = \kappa(x_i, x_j) \). Notationally, \( \{ f_x, x \in \mathcal{X} \} \sim GP(\mu, \kappa) \). It is called a zero mean process if \( \mu \equiv 0 \) and a stationary process if \( \kappa(x_i, x_j) \equiv \kappa(x_i - x_j) \) (that is, the covariance function is translation invariant).

Now, consider \( \mathcal{H} \) to be a RKHS of functions defined from \( \mathcal{X} \) to \( \mathbb{R} \) with a kernel \( \kappa \). Note that any such kernel is a legitimate covariance function. Then, a \( GP(\mu, \kappa) \)
behaves as a prior on the space $\mathcal{H}$. In other words a function $f_0 \in \mathcal{H}$ is considered as a realization, \( \{f_x(\omega_0), x \in \mathcal{X} \} \) (for a fixed $\omega_0 \in \Omega$), from a $GP(\mu, \kappa)$. Similarly, $f_1 \in \mathcal{H}$ is considered as another realization \( \{f_x(\omega_1), x \in \mathcal{X} \} \) (for another fixed $\omega_1 \in \Omega$), so on and so forth. Thus, depending on the mean function, $\mu$, and the covariance function $\kappa$, a $GP$ prior on $\mathcal{H}$, renders some functions in $\mathcal{H}$ “more probable” than the others. \cite{pillai} provides a functional theoretic foundation for using non-parametric prior specifications, such as Gaussian process in Bayesian kernel modeling.

Thus, for a typical (nonlinear) regression setup,

$$y = f(x) + \epsilon$$

where $\epsilon \sim \text{Gau}(0, \lambda_0)$ is the independent noise, a zero-mean GP prior on the underlying function amounts to the assumption that observations, $Y = (y_1, \ldots, y_n)'$, measured at $X = \{x_1, \ldots, x_n\}$, respectively, has the following distribution

$$Y \sim \text{Gau}_n(0, K + \lambda_0 I), \quad (1.5.1)$$

for all $n \in \mathbb{N}$ and all $\{x_i\}_{1}^{n} \in \mathcal{X}$.

As discussed in \cite{rasmussen}, Section 2.2, the best prediction (under squared error loss) at a new set of points, $X^v$, given by the conditional expectation, $E[Y^v|X, Y, X^v]$, turns out to be the same as that obtained from the ridge regression framework given by (1.2.5). And this is in fact a fallout of the representer theorem, Theorem 7.

Classically, the the parameters used to specify the mean and the covariance functions are treated as hyper-parameters and estimated using either the maximum likelihood method (MLE) or the maximum a posteriori method (MAP). Let $\theta$ represents all the parameters in distribution of (1.5.1) and $p(Y|X, \theta)$ represents the likelihood,
then the MLE method maximizes $p(Y|X, \theta)$ as a function of $\theta$; with a prior, $p(\theta)$, on $\theta$, the MAP method maximizes, $p(Y|X, \theta)p(\theta)$. In fact, as discussed in Rasmussen and Williams (2006, Section 5.4.1), one often just Sundarajan and Keerthi (2001) investigates, via simulation experiments, predictive approaches based on Geisser’s predictive sample reuse methodology and compares it with standard CV technique. These procedures, as discussed in Rasmussen and Williams (2006, Section 5.4.2) are akin to maximizing the leave-one-out log predictive probabilities

$$\sum_{i=1}^{n} \log p(y_i|X, Y_{-i}, \theta),$$

which is then optimized using conjugate gradient techniques.

1.6 Learning the Kernel and Model Selection

1.6.1 “Learning the kernel”

Given a problem, there may or may not exist any data generation process; more often than not in “real-life data” it does not. The goal of any estimation or prediction procedure is to come up with a model which not only does a “good” job of describing the current data but also, is guaranteed to perform “reasonably well” on any other (may be, new) “similar” set of data. So, whether or not there is a “true” process, from a modeling perspective, this amounts to the assumption that there is a “proxy” to the data generation process (Spiegelhalter et al. 2002, Sec. 2.2) which is representable by one among a class of processes and the mentioned goal is to find the “best” process from this class. Additionally, to account for the inherent uncertainty of any natural process an error term is included, which is often additive. Such a class of processes is generally either a class of parametric or non-parametric models.
So, what this RKHS theory does is to provide a unified mathematical framework for a vast collection of modeling strategies. Thus, any model (or the construed data generation process) is represented by a functional dependence (may be noisy) from the predictors to the response. The model construction in all these methodologies starts out by assuming that the unknown (either “true” or “proxy”) function belongs to a certain reproducing kernel Hilbert space.

In the RKHS setup, although the representer theorem (Theorem 7) provides a necessary condition for the minimizer function; from a “learning the kernel” point of view, it provides minimal information about the the underlying Hilbert space itself. As we know (Theorem 3), corresponding to a chosen kernel function, there is a unique RKHS. In other words, any particular choice of \( \kappa(\cdot, \cdot) \), completely characterizes its Hilbert space. In this sense, any problem of “learning the kernel” amounts to a problem of “learning the Hilbert space” itself!

Given a (training) data \( T_n = \{(y_i, x_i) : i = 1, \ldots, n\} \), and having chosen a kernel \( \kappa \), the resulting search space is no longer the entire \( \mathcal{H}_\kappa \), but rather the class of linear combinations of a finite collection of basis functions represented by \( \{\kappa(x_i, \cdot) : i = 1, \ldots, n\} \). What the representer theorem states is that, with an evaluation criterion given by the regularized cost function (loss + penalty), this is the best possible search space:

\[
\mathcal{H}_{\kappa, T_n} = \text{span}\{\kappa(x_i, \cdot) : i = 1, \ldots, n\}.
\]

Clearly, \( \mathcal{H}_{\kappa, T} \) is a proper subspace of \( \mathcal{H}_\kappa \). Thus, the problem of “learning the kernel” is a hugely ill-posed problem, where the goal is to identify \( \mathcal{H}_\kappa \), but all we are allowed
to search is, $\mathcal{H}_{\kappa,T_n}$! Even if the oracle says that the “true” function belongs to,

$$\mathcal{H}_{\kappa,O} = \text{span}\{\kappa(z_i, \cdot) : i = 1, \ldots, m\},$$

for some points $\{z_i\}_{i=1}^m \in \mathcal{X}$. It is very likely that the two subspaces, $\mathcal{H}_{\kappa,T_n}$ and $\mathcal{H}_{\kappa,O}$ are practically “disjoint.” If one believes in the parsimony of the nature, then the assumption, $\mathcal{H}_{\kappa,O} \subset \bigcup_{n=1}^{\infty} \mathcal{H}_{\kappa,T_n}$, can be justified in an asymptotic sense.

Thus, the literature on “learning the kernel” is geared towards picking one kernel from a bag of well-behaved pre-specified class of kernels. Such a class, is either finite, or generated by another finite class of kernels.

Let the regularized cost function be represented as,

$$C(f; \kappa, \lambda|\mathcal{T}) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \Omega(\|f\|_{\mathcal{H}_{\kappa,T}})$$

where $L(\cdot, \cdot)$ is the loss function and $\Omega(\cdot)$ is the penalty function. For brevity, the dependence of the cost function $C$ on the loss and the penalty is ignored, although they are intrinsically dependent. Let

$$C_\lambda(\kappa) := \inf \{C(f; \kappa, \lambda|\mathcal{T}) : f \in \mathcal{H}_{\kappa,T}\},$$

$$C_\lambda(\mathcal{K}) := \inf \{C_\lambda(\kappa) : \kappa \in \mathcal{K}\},$$

[Cristianini et al. (2001)] define the concept of kernel alignment between two kernels matrices, $K_1$ and $K_2$ (also see, Shawe-Taylor and Cristianini [2004], Defn. 3.20)

$$A(\mathcal{T}, K_1, K_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}}$$

where $\langle K_1, K_2 \rangle_F = \sum_{i,j=1}^{n} \kappa_1(x_i, x_j)\kappa_2(x_i, x_j)$ is the usual Euclidean inner-product in $\mathbb{R}^{n^2}$ and can also be seen as the absolute Frobenius norm of the Schür product.
They consider $K_2 = YY'$ and prove that the sample based estimate of this alignment metric is concentrated around its expected value and additionally for the Parzen’s window estimator, $\text{sign}(f(x)) = \text{sign} E_{(x',y')}(y' \kappa(x',x))$, the generalization error is suitably bounded above. As discussed later in Section 3.3, the tools investigated in this thesis is similar, in spirit, to the kernel-alignment technique.

A Bayesian approach, in a binary classification context, by treating the kernel entries as missing observations is proposed in Zhang et al. (2004) where they express the relationship between the kernel on the input space ($\mathcal{X} \times \mathcal{X} \to \mathbb{R}$) and a kernel on the output space ($\mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$) as a symmetric-definite generalized eigenproblem.

Another frequently used optimization technique in searching for an “optimal” kernel is the semidefinite programming which deals with the optimization of convex functions over the convex cone of symmetric positive semi-definite matrices. Lanckriet et al. (2004) provide a semidefinite programming technique to find an optimal linear combination of a pre-specified finite collection of kernels with a bounded trace. They define a generalized performance measure:

$$\omega_{c,\lambda}(K) = \max_\alpha 2 \alpha' 1 - \alpha' (G(K) + \lambda I) \alpha : 0 \leq \alpha \leq c, \alpha'y = 0,$$

with $\lambda \geq 0$ and $G_{ij}(K) = K_{ij}y_iy_j$, and provide methods to find an optimal $\kappa \in \mathcal{K}$ such that the trace($K$) is constant. Bach et al. (2004) claim that the above optimization problems are quite expensive and manageable for only small sample sizes and only for a limited choice of kernels; they provide a different optimization technique for the same problem. Another optimization approach given by Sonnenburg et al. (2006), rewrites the original problems as a semi-infinite linear program that can be efficiently solved by recycling the standard SVM. The benefit of this approach is that they
can generalize the formulation to a larger class of problems, including regression and one-class classification. A general drawback of learning the kernel matrix from such convex sets is that they overfit the data (Ong et al., 2005). But, some experimental analysis done by Saradhi and Karnick (2007) claim otherwise. Although they do support the other drawback in that the methods are unstable.

Inspired by the above works, Micchelli and Pontil (2005, Lem. 2, Thm. 7) provides a method to choose the best kernel among a class of convex set of kernels. Although the class may be uncountable, the optimal kernel is always a convex combination of at most \( n + 2 \) basic kernels. Ultimately, their methodology depends on prescribing a finite collection of kernels and then choosing the best one from the convex hull of this finite collection. An extension to this work, Micchelli and Pontil (2007), looks at a relationship between the problem of finding an optimal kernel and regularization in the dual space of the “space of continuous functions on a compact domain with values in a Hilbert space.”

In a very closely related work, Argyriou et al. (2005, Thms. 2,4, Cor. 1) considers \( \mathcal{K} \) to be a compact and convex set (with a generalization to compact Hausdorff space via a “meta-kernel”) and provide necessary and sufficient conditions to solve (1.6.1) (actually, they provide conditions for a saddle point for the general problem) and further claim that the optimal kernel has a finite representation with at most \( n + 1 \) terms. And when the class is a convex hull of finitely many basic kernels, the saddle point becomes a unique minimizer. The (minimax) cost function that they consider is given by,

\[
C_\lambda(\mathcal{K}) := - \max \left\{ \min \left\{ \lambda^{-1} \alpha' K \alpha + \alpha' \alpha + 4 y' \alpha : \alpha \in \mathbb{R}^n \right\} : \kappa \in \mathcal{K} \right\}.
\]
In a later article, Argyriou et al. (2006), this was generalized to the convex hull generated from a class of infinitely many basic kernels but with the restriction that the basic kernels are continuously parametrized. The resulting optimization problem is no longer convex and belongs to a larger class of “difference of convex” programs.

In all the methods described above, the regularization parameter, $\lambda$, is kept fixed. So, the “optimal kernel” is still subject to the choice of this parameter (which, almost always is done via cross-validation). In Johnson and Zhang (2008, Cor. 5.1), they use an “optimal” $\lambda$, to obtain a learning bound, which can be used to compare the performance of different kernels. Of course, there is no free lunch, they assume some prior knowledge of the range of the spectra of the kernels involved (see Sections V and VI there). In fact, Koo and R.M (2008), assume that the noise variance, $\lambda$, is known in order to search for the optimal kernel.

Some of the other works in this area include: Bousquet and Herrmann (2003) – in a SVM setup – consider a “spectral class” of kernels by varying the spectrum (but trace is constant) and keeping the eigenvectors fixed and provide bounds on the Rademacher complexities of the associated function classes. Ong et al. (2005) – in a SVM setup – define a RKHS over the space of kernels and provide an equivalent version of the representer theorem. The optimization problems are set in a semidefinite programming framework. Wu et al. (2007) – in binary classification setup – considers just a conglomeration of kernels, the associated function space (which is just the union of the RKHS of each kernel from the class) need not even be a a linear space and the optimization process there is analogous to partial likelihood. Lebanon (2006) – develops a metric from a parametric family that is based on maximizing the inverse volume of a given data set of points. de Diego et al. (2004) – combine kernels in a
simple way to build new kernels by quantifying, relative to the classification labels, the difference of information among the kernels.

A related problem of manifold learning has been discussed extensively in the literature over the past decade (Tenenbaum et al. 2000; Belkin and Niyogi 2003; Ye and Zhou 2008; Li et al. 2011). A novel approach, based on semidefinite programming, bridging the literature of manifold learning and kernel methods was proposed by Weinberger et al. (2004) where the kernel matrix was constructed by maximizing the variance in feature space subject to local constraints that preserve the angles and distances between nearest neighbors.

An interesting issue in the context of evaluating different kernels, is the size of their RKHS spaces. The theoretical tool (although not practically implementable) that has been studied in recent times is the covering numbers and packing numbers (Zhou 2002; Pontil 2003; Zhou 2003).

### 1.6.2 Model selection techniques

Given any data, with sufficient number of parameters, it is very easy to over fit. Hence, many of the the different model selection criteria can be thought of as “regularization” to avoid this over fitting. Of course, the concept of model selection is more than regularization, especially when distributional assumptions are being evaluated. But all most all regularization methods can be thought of as model selection strategies.

Starting from classical regression model selection methods, such as, forward selection, backward elimination, Mallows Cp, the Statistical literature is ripe with various techniques, a few of which are described below.
AIC was first proposed by [Akaike (1974)], and is given as

\[ AIC = -2 \log(\text{max } L) + 2p \]

where \( p \) is the number of parameters of the model and \( L \) is the likelihood given the data. Over the years many modifications have been proposed, such as, the following small sample correction,

\[ AIC = -2 \log(\text{max } L) + 2p + \frac{2p(p-1)}{n-p-1}. \]

A comprehensive information on AIC and other model selection strategies are given in [Burnham and Anderson (2002)].

[Takeuchi (1976)] generalizes the AIC by relaxing the assumptions that the true model belongs to the class of models being considered, although, doing so makes, TIC, hard to compute in practice.

Later, [Schwarz (1978)], proposed a modification to AIC, by finding an asymptotic Bayes solution under an exponential family,

\[ BIC = -2 \log(\text{max } L) + p \log n. \]

This is also known as Schwarz Criterion (or SBC or SBIC). Generalizations to the BIC criterion have been studied by [Berger et al. (2003)] and [Chakrabarti and Ghosh (2006)] among others.

A relatively new proposal (see [Spiegelhalter et al. 2002], Eqns. 10, 37) is the Deviance Information Criterion, DIC, given by

\[ DIC = 2E_{\theta|X} D(\theta) - D(E_{\theta|X}\theta), \]
where \( D(\theta) = -2 \log L(\theta) \) is the usual deviance statistics for the likelihood \( L \), and the expectation is under the posterior distribution. Instead of the posterior mean, one could alternatively use the posterior median or mode here.

Some of the other measures include, Minimum Description Length (MDL, Rissanen 1978), Generalized Information Criterion (GIC, Atkinson 1980), Fisher Information Criterion (FIC, Wei 1992), Network Information Criterion (NIC, Murata et al. 1994), Kullback Information Criterion (KIC, Cavanaugh 1999), Subspace Information Criterion (SIC, Sugiyama and Ogawa 2001) – a generalization of Mallows \( C_L \) when the function belongs to a Hilbert space, and the many variants of these.

In a regularized framework, leave-one-out cross-validation (CV) and generalized cross-validation (GCV) can also be considered as model selection techniques,

\[
CV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \hat{y}_{i\lambda})^2}{(1 - s_{ii}(\lambda))^2} \\
GCV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \hat{y}_{i\lambda})^2}{(1 - n^{-1} \text{tr} S(\lambda))^2}
\]  

(1.6.1)

where, \( S(\lambda) = [s_{ij}(\lambda)] \) is the smoother matrix, \( \hat{Y} = S(\lambda)Y \). As discussed in Hastie et al. (2001, pp. 216-217) GCV is an approximation to the leave-one-out CV for linear fitting under squared-error loss. A detailed work by Zhang (2003) provides bounds on expected leave-one-out cross-validation errors and its variance; which lead to generalization bounds for various kernel algorithms. In a recent work Braun et al. (2006) provides a way to select the regularization parameter based on a “cut-off” dimension of the spectrum of the gram matrix.

In addition, there are other variable selection algorithms, such as, LASSO, LARS, Bridge regression, COSSO, and Non-negative Garrote; model averaging methods such
as, Boosting and Bragging. Also, worth mentioning are the Structural Risk Minimization and VC-dimension criteria of Vapnik (2000).

A Bayesian approach to the concept of variable selection is the Relevance Vector Machine proposed by Tipping (2001). In a recent work Fokoué and Goel (2009) explored the RVM mechanism from the perspective of optimal statistical designs. The critical issue with the RVM technique is that the “optimal” design points that get selected is based on the chosen model kernel. So, the issue of variable selection is inherently confounded with the problem of model selection.
Chapter 2: Gaussian Process Regression Diagnostics

As discussed in Section 1.5, consider the regression setup,

\[ y = f(x) + \epsilon \]

with independent additive noise \( \epsilon \). We examine this regression problem in two frameworks: (a) Classical regularization (that is, constrained optimization) and (b) Gaussian Process Regression (GPR). Both these frameworks utilize a kernel function. In the case of regularization, \( f \) is assumed to belong to a RKHS, and the kernel function provides easy computations, through inner products, in its (possibly infinite dimensional) feature space whereas in the GPR case the kernel acts as the covariance function for the Gaussian Process prior on \( f \). We will denote this kernel function by \( \kappa(\cdot, \cdot; h) \), parameterized by \( h \). For the simulation experiments discussed in this Chapter \( \kappa \) was taken to be a RBF (Gaussian) kernel defined by,

\[ \kappa(x_1, x_2; h) = \exp \left\{ -\frac{1}{h} \| x_1 - x_2 \|^2 \right\}. \]

It should be noted that, although the simulation studies done here were restricted to Gaussian kernels, the algebraic results discussed in Section 3.3 hold for more generic kernels. The essential requirement for the algebraic results is a majorization relationship between the eigenvalue vector of the Gram matrix corresponding to the “true”
kernel and that of the model kernel. The concept of a “true” kernel and a model kernel will be explained in the following sections. Based on the direction of this majorization relationship interesting diagnostic concepts emerge. A class of (normalized) kernels closed under the Hadamard product (see Section 3.1.1) will always satisfy such majorization requirements. In particular, the family of Gaussian kernels is just a specific example of this generic family of kernels. Further, as briefly discussed through Result 2 (Section 3.1.2), some of the results from Section 3.3 extends to convex combination of kernels as well.

Suppose the underlying process, \( f \), is observed at the training points, \( X^{tr} = \{X^{tr}_1, \ldots, X^{tr}_n\} \) and is to be predicted at the validation points \( X^v = \{X^v_1, \ldots, X^v_m\} \) where \( X^{tr}_i, X^v_j \in \mathcal{X} \) for all \( i, j \). The domain, \( \mathcal{X} \subset \mathbb{R}^d \). Let

\[
K_h = [\kappa(X^{tr}_i, X^{tr}_j; h)]_{n \times n} \quad \text{and} \quad K^v_h = [\kappa(X^v_i, X^v_j; h)]_{m \times m}
\]

denote the Gram matrices at the training points and validation points, respectively, and

\[
G_h = [\kappa(X^v_i, X^{tr}_j; h)]_{m \times n}
\]

denote the similarities between the training and validation points. Also, let the unobserved functional values at these two sets of domain points be denoted by \( f^{tr} = (f(X^{tr}_1), \ldots, f(X^{tr}_n))' \) and \( f^v = (f(X^v_1), \ldots, f(X^v_m))' \) and the observed (noisy) measurement of \( f^{tr} \) be denoted by \( Y^{tr} \).

In the next section we briefly review the similarities between the classical regularization and the GP frameworks for regression analysis. In the following section we explore four ways, borrowing concepts from both the frameworks, to measure the
“closeness” between underlying process (characterized by a “true” kernel) that generates the data and the modeling process (characterized by a “training” kernel) that is used for predictions on future test points. In Section 2.3 we restrict to two of the four ways, and compare them to their expectations under the modeling distribution. Based on the observations made thus far, we develop two diagnostic tools in Section 2.4.

2.1 GPR and Classical Regularization: a Review

Classical Regularization Problem:

The classical optimization problem described by (1.2.1) using squared error loss, 
\[ L(y_i, f(x_i)) = (y_i - f(x_i))^2, \]
and RKHS functional norm penalty, \( \Omega(\|f\|_{\mathcal{H}_{\kappa_h}}) = \|f\|_{\mathcal{H}_{\kappa_h}}^2, \)
is
\[
\min_{f \in \mathcal{H}_{\kappa}} \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(X_{tr}^i))^2 + \lambda \|f\|_{\mathcal{H}_{\kappa_h}}^2.
\]

As stated by the Generalized Representer Theorem (Theorem 7), the optimal solution is a linear combination of the kernel basis functions:

\[
f(\cdot) = \sum_{j=1}^{n} \kappa(\cdot, X_{tr}^j; h) \alpha_i,
\]
where, \( \alpha = (\alpha_1, \ldots, \alpha_n)' \) are the unknown coefficients. It is easy to check that for this \( f, \)
\[
\|f\|_{\mathcal{H}_{\kappa_h}}^2 = \left( \sum_{j=1}^{n} \kappa(\cdot, X_{tr}^j; h) \alpha_i, \sum_{j=1}^{n} \kappa(\cdot, X_{tr}^j; h) \alpha_i \right) = \alpha' K_h \alpha.
\]
Thus, the above optimization problem translates to (similar to the Example 3 of Section 1.2)
\[
\min_{\alpha} (Y_{tr} - K_h \alpha)'(Y_{tr} - K_h \alpha) + \lambda \alpha' K_h \alpha.
\] (2.1.1)
The optimal solution to (2.1.1) is then given by

\[ \hat{\alpha} = (K_h + \lambda I)^{-1} Y^{tr}, \]

and the estimated function under model bandwidth \( h \) is given by

\[ \hat{f}_h(\cdot) = \sum_{i=1}^{n} \kappa(\cdot, X^{tr}_i; h) \hat{\alpha}_i. \]  \hspace{1cm} (2.1.2)

In particular, the in-sample estimates are

\[ \hat{Y}^{tr} = K_h \hat{\alpha} = K_h (K_h + \lambda I)^{-1} Y^{tr} = A_{h,\lambda} Y^{tr}, \]

where,

\[ A_{h,\lambda} = (K_h + \lambda I)^{-1} K_h \] \hspace{1cm} (2.1.3)

and the out-of-sample predictions are

\[ \hat{f}^v = G_h (K_h + \lambda I)^{-1} Y^{tr}. \]

The penalty coefficient, \( \lambda \), is typically chosen using some cross-validation techniques. The discussion in Section 3.4.1 argues that in this framework, we can improve the search process for an “optimal” \( \lambda \) by measuring how much the measurements, \( Y^{tr} \), can be decorrelated using the training Gram matrix.

Further, as discussed below the estimates, \( \hat{Y}^{tr} \), and the predictions, \( \hat{f}^v \), are the same as those obtained via GPR. This fact has been widely noted in the literature, see for example, Schölkopf and Smola (2002, Section. 16.3.4) and Rasmussen and Williams (2006, Eqns. 2.25 and 2.35).

It is worth noting here that this regularization framework does not have the concept of an underlying generation process. The Generalized Representer Theorem only
states that if we want to find the optimal function for prediction, given the training data, from the entire RKHS, \( \mathcal{H}_{\kappa h} \), then it suffices to search only within the linear combinations of the kernel basis functions. In cases, where the data is indeed generated from a specific \( \mathcal{H}_{\kappa h_0} \), this regularization framework does not have any mechanism to quantify the cost of using \( \mathcal{H}_{\kappa h} \) instead of \( \mathcal{H}_{\kappa h_0} \).

**GPR via predictive distribution:**

Now, assume a GP prior on \( f \) with a covariance kernel \( \kappa(\cdot, \cdot; h_0) \) where \( h_0 \) is termed as the “true” bandwidth. Following (1.5.1), consider the joint generation process,

\[
\begin{pmatrix} f^{tr} \\ f^v \end{pmatrix} \sim \text{Gau}_{n+m} \left( 0, \begin{pmatrix} K_{h_0} & G'_{h_0} \\ G_{h_0} & K_{h_0}^v \end{pmatrix} \right).
\]

(2.1.4)

In this respect, \( \kappa(\cdot, \cdot; h_0) \) is the “true” kernel corresponding to the underlying generation process. Consider independent additive noise

\[
\epsilon \sim \text{Gau}(0, \lambda_0),
\]

(2.1.5)

where, \( \lambda_0 \) is termed as the “true” noise variance. In general, if one considers a scalar multiple, \( \sigma_f^2 \), for the variance terms in (2.1.4) and \( \epsilon \sim \text{Gau}(0, \sigma^2) \), then \( \lambda_0 \) is regarded the “true” signal to noise ratio, \( \sigma^2 / \sigma_f^2 \). With, \( Y^{tr} = f^{tr} + \epsilon \), the marginal distribution of the measurements is given by

\[
Y_{n \times 1}^{tr} \sim \text{Gau}_n \left( 0, K_{h_0} + \lambda_0 I \right),
\]

and the predictive distribution of the function values at the validation points is thus given by,

\[
f^v \mid X^{tr}, Y^{tr}, X^v \sim \text{Gau}(\mu(f^v), \text{cov}(f^v))
\]

(2.1.6)
where

\[
\mu(f^v) := E \{ f^v | X^{tr}, Y^{tr}, X^v \} = G_{h_0}(K_{h_0} + \lambda_0 I)^{-1} Y^{tr},
\]

\[
\text{cov}(f^v) := V \{ f^v | X^{tr}, Y^{tr}, X^v \} = G_{h_0}(K_{h_0} + \lambda_0 I)^{-1} G_{h_0}.
\]

In particular, for \( X^v = X^{tr} \), we get the same in-sample estimates as earlier,

\[
\hat{Y}^{tr} = K_{h_0} (K_{h_0} + \lambda_0 I)^{-1} Y^{tr}.
\]

It is important to clarify the notation a bit here. In all practical cases we will not know \( h_0 \) and \( \lambda_0 \). We assume that the distributions (2.1.4) and (2.1.5) hold for \( h \) and \( \lambda \), respectively, which we use to fit the training model. Thus, \( h \) and \( \lambda \) will be termed as the hyper-parameters corresponding to the modeling process.

To evaluate the quality of these estimated models as well as to evaluate the appropriateness of the chosen (model) kernel for the data, we borrow four concepts from the two frameworks: (a) the Generalized Cross Validation (GCV) for the prediction process, (b) the functional norm of the estimated function, (c) the regularization cost for the optimal solution, and (d) the Mean Squared Error for the out-of-sample predictions. These are described in the next section where we explore their behavior across various choices of \( h \) and \( \lambda \).

### 2.2 Preliminary Experiments

In this section we discuss the four concepts mentioned above through computer simulations. First we provide their expressions; then we discuss the simulation experiments in the subsections that follow.
2.2.1 Expressions for GCV, Functional Norm, Optimal Cost, and Out-of-sample Predictions

GCV, functional norm, and the optimal cost:

As discussed in Section 1.6.2, GCV can be regarded as a model selection criterion and in this context, as defined in (1.6.1), it is given by

\[
GCV(h, \lambda) = \frac{n^{-1}Y^{tr}(I - A_{h,\lambda}')^2Y^{tr}}{(n^{-1}\text{tr}(I - A_{h,\lambda}'))^2} = \frac{nY^{tr}(K_h + \lambda I)^{-2}Y^{tr}}{\text{tr}(K_h + \lambda I)^{-1}}
\]

\[
= \text{tr} \left\{ B_{h,\lambda} (K_h + \lambda I)^{-1} D_{Y^{tr}} \right\}, \quad (2.2.1)
\]

where, \( I - A_{h,\lambda}' = \lambda (K_h + \lambda I)^{-1} \) for \( A_{h,\lambda} \) is defined in (2.1.3) and

\[
B_{h,\lambda} = n \left( \text{tr}(K_h + \lambda I)^{-1} \right)^{-2} (K_h + \lambda I)^{-1}
\]

and \( D_{Y^{tr}} = Y^{tr}Y^{tr}' \) is the “data-kernel.” In addition, the usual leave-one-out cross-validation can be expressed in a similar form:

Note that the estimated function \( \hat{f}_h \), by construction, belongs to the Hilbert space corresponding to the kernel \( \kappa(\cdot, \cdot; h) \). Thus, its functional norm is given by,

\[
\nu(h, \lambda) := \| \hat{f}_h \|_{H_\kappa} = \hat{\alpha}'K_h\hat{\alpha}
\]

\[
= Y^{tr}(K_h + \lambda I)^{-1}K_h(K_h + \lambda I)^{-1}Y^{tr},
\]

\[
= \text{tr} \left\{ A_{h,\lambda} (K_h + \lambda I)^{-1} D_{Y^{tr}} \right\}, \quad (2.2.2)
\]

and the regularization cost corresponding to the optimal solution of (2.1.1) is given by

\[
C(h, \lambda) := (Y^{tr} - K_h\hat{\alpha})'(Y^{tr} - K_h\hat{\alpha}) + \lambda \hat{\alpha}'K_h\hat{\alpha}
\]

\[
= Y^{tr}(I - A_{h,\lambda}')^2Y^{tr} + \lambda \nu(h, \lambda).
\]
We will call this as the “optimal cost.”

As will be seen later in Section 3.3, expressing the GCV and the functional norm in the forms of (2.2.1) and (2.2.2) is particularly interesting.

“Out-of-sample” predictions (validation set MSE):

Measurements on the validation set $X^v$ can be used to test the estimated model. Although, these new domain points are independent of the training points, $X^{tr}$, the measurements at these points are not independent of the training measurements. In the context of GPR, to ensure that both training and validation observations correspond to the same realization of the process, they have to be generated from a single joint distribution, as shown in (2.1.4). In other words, the data generation process can be thought of as a single experiment, a part of which is used to train the model and the remaining part is used to test/validate the estimated model. Let $Y^v_{m \times 1}$ denote the measurements at the validation points. These are generated along with $Y^{tr}$ as follows,

$$
\left( Y_{n \times 1}^{tr}, Y_{m \times 1}^v \right) \sim \text{Gau}_{n+m} \left( 0, \begin{pmatrix} K_{h_0} + \lambda_0 I_n & G_{h_0} \\ G_{h_0}^t & K_{h_0}^v + \lambda_0 I_m \end{pmatrix} \right). \tag{2.2.4}
$$

The model is then trained using a bandwidth $h$, and penalty $\lambda$. The prediction at these validation points follows from (2.1.2) or (2.1.6),

$$
\hat{Y}^v = G_h (K_h + \lambda I)^{-1} Y^{tr},
$$

and the mean squared error for the validation set is given by,

$$
\text{MSE}_v(h, \lambda) := m^{-1} (Y^v - \hat{Y}^v)'(Y^v - \hat{Y}^v). \tag{2.2.5}
$$
2.2.2 Setup for simulation experiments

We want to study the four entities described earlier from the perspective of both the training model and the “true” underlying generation process. The approach taken is the following. We start out with a few underlying generators. For each such generation process, we fit a number of training models. The issues that we want to study, and hopefully address, are summarized as follows:

- Given the the underlying generation process, are some training models “better suited” (“optimal”) to the data than the others?
- How does such an “optimal” training model, if exists, change as we change the underlying generation process itself?
- If applicable, how do each of these entities change as we move away from an “optimal” training model to a sub-optimal one?

The domain points were generated uniformly from a $d = 4$ dimensional space: for the training purpose $X_{tr1}, \ldots, X_{trn} \sim U(-4,4)^d$. Independently, for validation purpose, $X_{v1}, \ldots, X_{vm} \sim U(-4,4)^d$ were generated. For the analyses discussed here, $n = 200$ training points and $m = 500$ validation points were used. The distribution of the squared Euclidean distances of the $n = 200$ domain points are:

<table>
<thead>
<tr>
<th>$|x_i - x_j|^2$</th>
<th>Min 10% 25% 50% 75% 90% Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>12.3 22.3 37.9 56.9 76.0 183.4</td>
</tr>
</tbody>
</table>

In our context, the underlying generation process is characterized by two hyper-parameters: $h_0$, the bandwidth corresponding to the underlying kernel, and $\lambda_0$, the
noise variance. Since, we are interested in the problem of “learning the kernel,” \( h_0 \) is our parameter of interest. Eight different “true” bandwidths, equidistant between 3 and 30 were considered: \( h_0 \in \{3, 6.9, 10.7, 14.6, 18.4, 22.3, 26.1, 30\} \), for the simulations. Of these, analyses corresponding to only \( h_0 \in \{3, 14.6, 26.1\} \) are shown in this thesis and they are representative of the eight values studied. We know that using a large bandwidth makes the generating function very smooth and thereby uninformative. In that respect the above choices of \( h_0 \) values are biased towards the low end. The “true” noise variance was fixed at \( \lambda_0 = 0.01 \). Later, in Section 3.3.5 we very briefly touch upon the issue of varying \( \lambda_0 \) along with varying the training set points.

For training the model wide ranging values of \( h \) and \( \lambda \) were considered. To better visualize the estimated models two different sets of \( h \) values were used. In the first set, the same values of \( h \) were used for all the different choices of \( h_0 \): 50 points, equidistant on log-scale, were chosen from the interval \([1.5, 60]\). In the second set, \( h \) were chosen to be around \( h_0 \): 50 points, equidistant on log-scale, were chosen from the interval \([h_0/2, 2h_0]\). While the former provided a view of the large-scale behavior, the later “zoomed” in on the behavior near \( h_0 \). For the model’s noise variance, \( \lambda \): 50 points, equidistant on log-scale, were chosen from \([e^{-10}, 1]\).

To understand the variability in these entities due to the inherent randomness in the measurements, we generated \( N = 1000 \) i.i.d replications following (2.2.4), denoted as \((Y^{tr}_{i}, Y^{v}_{i})'\) for \( i = 1, \ldots, N \). In what follows, we show the figures using \( Y^{tr}_{1} \) in the main text and refer to the figures using \( Y^{tr}_{2}, \ldots, Y^{tr}_{10} \) in the Appendices. These figures based on ten out of the 1000 replications should provide an idea of both the variability and consistency of the observed patterns.

We start out by exploring the patterns on \( h \times \lambda \) grid in the following subsection.
2.2.3 Investigations on $h \times \lambda$ Grid

Let us consider GCV given by (2.2.1) first. Figure 2.1 shows GCV($h, \lambda$), using $Y_{\text{tr}}^r$, over a grid of $h \times \lambda$ values. The horizontal and vertical axes are log($h$) and log($\lambda$) respectively. The top row shows the plots for the large range of $h \in [1.5, 60]$ and the bottom row shows the same plots but “zoomed” in near $h_0$, $h \in [h_0/2, 2h_0]$. The corresponding $h_0$ values are specified on top of each column.

We see a gradual increase in the GCV values (that is, deterioration of the quality of the estimated model) as move away from $(h_0, \lambda_0)$. Interestingly, the marginal
behavior for $\lambda < \lambda_0$ seems to be different from that of $\lambda > \lambda_0$. Defining $h^*$ as

$$h^*(\lambda) = \arg\min_h \text{GCV}(h, \lambda),$$

we observe, based purely on visual inspection, that for $\lambda < \lambda_0$, $h^*(\lambda)$ tends to larger than $h_0$ and for $\lambda > \lambda_0$, $h^*(\lambda)$ tends to be smaller than $h_0$. This is more evident for larger values of $h_0$. In other words, modeling noise using a prior with lower variance (than the “true” noise variance) biases the estimation procedure towards a more smooth model, that is “away from the data,” and using a prior with more variance (than the “true” noise variance) biases the procedure towards a less smooth model, that is, “closer to the data.” Later, in Section 2.4 where we discuss the actual diagnostic tools, we will see a similar, but more distinctly visible, asymmetric pattern (e.g., see Figures 3.1 – 3.4). Of course, the global minimum on the grid is still near $(h_0, \lambda_0)$ and in particular, at $\lambda = \lambda_0$ the minimum GCV seems to be always attained near $h_0$, for all the values of $h_0$ considered. Similar patterns were observed across multiple replications of the measurements, $Y_{tr}^1, \ldots, Y_{tr}^{10}$, which are shown in Figures A.1 – A.9, respectively.

Next, let us consider the functional norm given by (2.2.2). Figure 2.2 shows $\log \nu(h, \lambda)$, using $Y_{tr}^1$, on a $h \times \lambda$ grid. The attributes are same as that of Figure 2.1.

Unlike the patterns for the GCV plot, we do not observe any similar behavior near $(h_0, \lambda_0)$. Although, it is not surprising that marginally, for a fixed $h$, larger values of $\lambda$ imposes larger penalty on the smoothness of the estimated function, hence yields smaller norms of the estimated function (that is, smoother functions). The other marginal is more interesting: for a fixed $\lambda$, the minimum norm seems to be attained at a bandwidth $h^*$ generally below $h_0$, yet in small neighborhoods around $h_0$ the functional norm seems monotonic. Again, similar patterns were observed for
Figure 2.2: Functional norm (2.2.2): $\log \nu (h, \lambda)$ on a $h \times \lambda$ grid using $Y_1^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).

multiple replications of the measurements, $Y_2^{tr}, \ldots, Y_{10}^{tr}$ which are shown in A.10 – A.18 respectively.

The other two measures considered initially were the regularization cost for the optimal solution, $C(h, \lambda)$, given by (2.2.3) and the mean squared error of the predictions on the validation set, $\text{MSE}_v(h, \lambda)$, given by (2.2.5). Figure A.19 shows $C(h, \lambda)$ using $Y_1^{tr}$ on a $h \times \lambda$ grid. It does not seem too informative other than the obvious fact that marginally, for a fixed $h$, as $\lambda \to 0$, the training model tends to overfit the data and hence drives $C(h, \lambda) \to 0$. Figure A.20 shows $\text{MSE}_v(h, \lambda)$ using $Y_1^{tr}$ and its pattern is similar to GCV$(h, \lambda)$. Such patterns, for both $C(h, \lambda)$ and $\text{MSE}_v(h, \lambda)$, were consistently seen using repeated measurements and so in the context of the current thesis, both $C(h, \lambda)$ and $\text{MSE}_v(h, \lambda)$ were not investigated any further.
2.2.4 Investigations at $\lambda = \lambda_0$

As seen from Figures 2.1 and 2.2 (along with Figures A.1 – A.18), and the accompanied discussions, the variability and sensitivity of both GCV and functional norm across the entire $h \times \lambda$ grid is quite considerable. To get a better handle on the behavior of these tools we restrict to the cross-section at $\lambda = \lambda_0$ and investigate further. In other words, in what follows we assume that the “true” noise variance is known.

$\begin{align*}
\text{GCV at } \lambda = \lambda_0: & \quad \text{GCV}(h, \lambda_0) \text{ as a function of } \log(h) \text{ using } Y_{1}^{tr}. \\
\text{Top row: } 1.5 \leq h \leq 60; & \quad \text{bottom row: } h_0/2 \leq h \leq 2h_0. \text{ Dotted line is at } \log(h_0).
\end{align*}$

Let us consider the GCV first. Figure 2.3 shows GCV$(h, \lambda_0)$ using $Y_{1}^{tr}$ as a function of $\log(h)$. Very distinctly, these curves attain a minimum near $h_0$. See Figures A.21 – A.29 for replications using measurements $Y_{2}^{tr}, \ldots, Y_{10}^{tr}$. Further, within suitably small neighborhoods around $h_0$, as a function of $\log(h)$, the behavior seems
to be quite symmetric about log($h_0$) and monotonic on either side. The fact that the behavior of GCV is pretty similar to that of the MSE, (see Figure A.30), suggests that diagnostics based on just the training set might be sufficient, at least initially, to curtail the search space for a better fitting kernel.

Figure 2.4: Functional norm (2.2.2) at $\lambda = \lambda_0$: log $\nu (h, \lambda_0)$ as a function of log($h$) using $Y_{i}^{tr}$. Top row: $1.5 < h \leq 60$; bottom row: $h_0/2 < h \leq 2h_0$. Dotted line is at log($h_0$) (vertical) and log($n$) $\approx$ 5.3 (horizontal).

Now, let us consider the functional norm. Figure 2.4 shows log($\nu(h, \lambda_0)$) using $Y_{i}^{tr}$ as a function of log($h$). The horizontal dotted line is drawn at log($n$) = log(200) $\approx$ 5.3. As initially noted in the discussion for Figure 2.2, the function with the minimum functional norm seems to be consistently observed at $h^* < h_0$, for every $h_0$ considered in the simulations (even those not shown in these figures). (See Figures A.31 -- A.39 for the behavior using measurements $Y_{2}^{tr}, \ldots, Y_{10}^{tr}$.) Although this pattern could not be explained analytically, one can see why this happens, at least subjectively. Clearly,
\( \hat{f}_{h^*} \) is less smooth than \( \hat{f}_{h_0} \), but given the finite number of domain points, the smaller the bandwidth used for modeling the better is the training sample fit. One would expect that asymptotically, as \( n \to \infty \), that is, with more and more observation points, \( h^* \to h_0 \). Such asymptotic analyses was not done as part of this work.

Further, as discussed in Result 7 (see Section 3.3), \( \log(E_{h_0, \lambda_0} \nu(h_0, \lambda_0)) \) is bounded above by \( \log(n) \). And we observe from the Figures 2.4 and A.31–A.39 that, generally, when \( h \gg h_0 \), \( \log(\nu(h, \lambda_0)) > \log(n) \). So diagnostically, a model bandwidth \( h \) which results in \( \log(\nu(h, \lambda_0)) \) “considerably exceeding” \( \log(n) \), can be inferred to be not near the “true” \( h_0 \).

2.3 If model was the truth...

In this section we investigate the GCV and the functional norm under the distribution of the modeling process. In other words, we assume that the modeling distribution, characterized by \( h \) and \( \lambda \) is itself the underlying underlying generating distribution, that is, \( h_0 = h \) and \( \lambda_0 = \lambda \), and then try to understand the resulting patterns by comparing them with the patterns observed in the previous sections.

Under the modeling process, if we assume that (2.2.4) and (2.1.5) hold for \( h_0 = h \) and \( \lambda_0 = \lambda \), then,

\[
E_{h, \lambda} D_{Y \text{tr}} = K_h + \lambda I.
\]

This leads to the expected values of GCV and functional norm as given below:

\[
E_{h, \lambda} \text{GCV}(h, \lambda) = \text{tr} \{ B_{h, \lambda} \} = n \left( \text{tr} \{(K_h + \lambda I)^{-1}\}^{-1} \right), \quad (2.3.1)
\]

\[
E_{h, \lambda} \nu(h, \lambda) = \text{tr} \{ A_{h, \lambda} \} = \text{tr} \{(K_h + \lambda I)^{-1}K_h\}. \quad (2.3.2)
\]

Note that with the above assumption, these are computable from the data.
As discussed later in Section 3.3.1, for a fixed $\lambda$, as a function of $h$, both $\text{tr} \{ B_{h,\lambda} \}$ and $\text{tr} \{ A_{h,\lambda} \}$ as monotonically decreasing functions.

Figure 2.5: Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of log($h$) using $Y^{tr}_t$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$).

As discussed in the previous sections, we are interested in the behaviors at $\lambda = \lambda_0$, that is,

\[ E_{h,\lambda_0}GCV(h, \lambda_0) = \text{tr} \{ B_{h,\lambda_0} \} = n \left( \text{tr} \left\{ (K_h + \lambda_0 I)^{-1} \right\} \right)^{-1}, \]

\[ E_{h,\lambda_0}\nu(h, \lambda_0) = \text{tr} \{ A_{h,\lambda_0} \} = \text{tr} \left\{ (K_h + \lambda_0 I)^{-1} K_h \right\}. \]

We explore how these behave in comparison to the GCV and Functional norm computed using the training measurements.
Let us consider the GCV first. Figure 2.5 shows GCV\((h, \lambda_0)\) (denoted by the solid blue curve) using \(Y_1^{tr}\) alongside \(E_{h,\lambda_0}GCV(h, \lambda_0)\) (denoted by the dashed red curve) as a function of \(\log(h)\). Similarly for the functional norm, Figure 2.6 shows \(\nu(h, \lambda_0)\) (solid blue curve) using \(Y_1^{tr}\) alongside \(E_{h,\lambda_0}\nu(h, \lambda_0)\) (dashed red curve) as a function of \(\log(h)\). Corresponding figures using the measurements, \(Y_2^{tr}, \ldots, Y_{14}^{tr}\), are shown in Figures B.1 – B.9 for the GCV and Figures B.10 – B.18 for the functional norm.

![Figure 2.5](image1.png)

![Figure 2.6](image2.png)

Figure 2.6: Comparing \(\nu(h, \lambda_0)\) (solid blue curve) and \(E_{h,\lambda_0}\nu(h, \lambda_0)\) (dashed red curve) as a function of \(\log(h)\) using \(Y_1^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at \(\log(h_0)\).

Perusing though all these figures we see a very interesting pattern emerging for both GCV and functional norm. The two curves (solid blue and the red dashed), “cross” near the “true” bandwidth, \(h_0\). Of course, due to the inherent randomness in
the measurements, the crossing can occur on either side of $h_0$, but it is always “near” $h_0$.

These observations suggest that the difference of the two curves (for both GCV and functional norm) are important form the point of view of “learning the kernel.” In other words, considering the difference of the GCV (and functional norm) from its expectation under the modeling distribution, especially, the sign of these differences, have some information regarding the “true” underlying bandwidth, $h_0$.

We explore and investigate these differences in the next section.

2.4 Constructing the Diagnostic Tools

Based on the “crossing” phenomenon observed in the previous section, we construct the two main “diagnostic tools.” These tools are examined through their behavior across multiple repetitions using different training measurements in this section.

2.4.1 Expressions for the Tools

We want to study the behavior of the following differences:

Based on GCV,

$$T(h, \lambda) := \Gamma(h, \lambda) = \text{GCV}(h, \lambda) - E_{h,\lambda}\text{GCV}(h, \lambda)$$

$$= \text{tr} \left\{B_{h,\lambda} \left((K_h + \lambda I)^{-1} D_{Ytr} - I\right)\right\}. \quad (2.4.1)$$

and based on functional norm,

$$T(h, \lambda) := \Upsilon(h, \lambda) = \nu(h, \lambda) - E_{h,\lambda}\nu(h, \lambda)$$

$$= \text{tr} \left\{A_{h,\lambda} \left((K_h + \lambda I)^{-1} D_{Ytr} - I\right)\right\}. \quad (2.4.2)$$
We call $\Gamma(h, \lambda)$ as the “diagnostic tool based on GCV” and $\Upsilon(h, \lambda)$ as the “diagnostic tool based on functional norm.” Comparing the two expressions, we see that the matrix

$$
\Xi(\kappa_1, \lambda) := (K_h + \lambda I)^{-1} D_{Y_{tr}} - I
$$

is common to both and $\Gamma(h, \lambda)$ and $\Upsilon(h, \lambda)$ are just traces of $\Xi(\kappa_1, \lambda)$ pre-multiplied by $B_{h,\lambda}$ and $A_{h,\lambda}$ respectively. As we will see later in Section 3.4.1 that writing the above expressions in terms of $\Xi(\kappa_1, \lambda)$ is particularly interesting. We will call it the “decorrelation-discrepancy” matrix and discuss it further in Section 3.4.1.

In the next two subsections we explore these two tools, first on the $h \times \lambda$ grid, then focus on $\lambda = \lambda_0$.

As an aside, note that, in most practical application, we have finite number of measurements from only one realization of the underlying process. But, when possible, if we have (finite) measurements on multiple realizations (as we have been exploring through $Y_{tr}^{2}, \ldots, Y_{tr}^{10}$) of the underlying process, then we can define the diagnostic tools as

$$
\Gamma(h, \lambda) = \text{tr} \left\{ B_{h,\lambda} \left( (K_h + \lambda I)^{-1} S - N^{-1} I \right) \right\},
$$

$$
\Upsilon(h, \lambda) = \text{tr} \left\{ A_{h,\lambda} \left( (K_h + \lambda I)^{-1} S - N^{-1} I \right) \right\},
$$

where $S = N^{-1} \sum_{i=1}^{N} D_{Y_{tr}^i}$ is the sample dispersion matrix.

### 2.4.2 Geometric Interpretation of the Tools

Note that, for two real symmetric matrices $U$ and $V$, $\langle U, V \rangle := \text{tr}(UV)$ is an inner product in the vector space of $n \times n$ symmetric matrices as well as in the $\mathbb{R}^{n^2}$ Euclidean space. In this respect, the two diagnostic tools, at $\lambda = \lambda_0$, can be written
as,
\[
\Gamma(h, \lambda_0) = c \text{tr} \left\{ (K_h + \lambda_0 I)^{-2} D_{Y^tr} \right\} - \text{tr} (K_h + \lambda_0 I)^{-1}
\]
\[
= c \text{tr} \left\{ (K_h + \lambda_0 I)^{-2} (D_{Y^tr} - (K_h + \lambda_0 I)) \right\}
\]
and
\[
\Upsilon(h, \lambda_0) = \text{tr} \left\{ (K_h + \lambda_0 I)^{-1} K_h (K_h + \lambda_0 I)^{-2} D_{Y^tr} \right\} - \text{tr} \left\{ (K_h + \lambda_0 I)^{-1} K_h \right\}
\]
\[
= \text{tr} \left\{ (K_h + \lambda_0 I)^{-1} K_h (K_h + \lambda_0 I)^{-1} (D_{Y^tr} - (K_h + \lambda_0 I)) \right\}
\]
where \(c\) is the positive scalar in \(B_{h,\lambda_0}\).

These can now be interpreted as an “angle” between the model kernel, \((K_h + \lambda_0 I)\), and the data kernel, \(D_{Y^tr} = Y^{tr}Y^{tr'}\). The “angle” in this context is being measured with respect to certain multiplier matrices: \((K_1 + \lambda_0 I)^{-2}\) in the case of GCV based tool and \((K_h + \lambda_0 I)^{-1} K_h (K_h + \lambda_0 I)^{-1}\) in the case of functional norm based tool.

### 2.4.3 Diagnostic tools on \(h \times \lambda\) Grid

Let us consider, the tool based on GCV first. Figure 2.7 shows \(\Gamma(h, \lambda)\), given by (2.4.1), using the measurement \(Y_1^{tr}\) on a \(h \times \lambda\) grid. The attributes of this figure is similar to the other \(h \times \lambda\) plots seen earlier. The red pixels denote \(\Gamma(h, \lambda) < 0\) and the blue pixels denote \(\Gamma(h, \lambda) > 0\) with darker shades denoting larger values in magnitude. The loci of the sign-change points are (approximately) denoted by the solid curve cutting across the grid; in other words, the curve (approximately) denotes the roots of the equation
\[
\Gamma(h, \lambda) = 0.
\]

\(\Gamma(h, \lambda)\) using other measurements, \(Y_2^{tr}, \ldots, Y_{10}^{tr}\) are shown in Figures C.1 – C.9
Similarly, Figure 2.7 shows the diagnostic tool based on the functional norm, $\mathcal{Y}(h, \lambda)$, given by (2.4.2) using the measurement $Y_{1tr}^r$. The solid curve through the grid is an approximation for the loci of the roots of the equation

$$\mathcal{Y}(h, \lambda) = 0.$$ 

$\mathcal{Y}(h, \lambda)$ using other measurements, $Y_{2tr}, \ldots, Y_{10tr}$ are shown in Figures C.10 – C.18.

![Figure 2.7: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{1tr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). Solid curve (approximately) denotes the sign change boundary.](image)

There is clearly a sign-change pattern visible from these plots. This pattern is certainly asymmetric with respect to $\lambda$. In the sense that, for $\lambda < \lambda_0$ the sign change seems to occur near a $h^* < h_0$ and for $\lambda > \lambda_0$ the it seems to occur near a $h^* > h_0$. Further as $\lambda$ get closer to $\lambda_0$ this sign-change point also seems to get closer to $h_0$. We investigate this marginal behavior at $\lambda_0$ next.
Figure 2.8: Diagnostic tool based on functional norm (2.4.2): \( Y(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_1^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary.

### 2.4.4 Diagnostic tools at \( \lambda = \lambda_0 \)

We restrict to the marginal behavior at \( \lambda = \lambda_0 \) here. Figure 2.9 shows \( \Gamma(h, \lambda_0) \) and Figure 2.10 shows \( \Upsilon(h, \lambda_0) \) using the measurement \( Y_1^{tr} \) as a function of \( \log(h) \). Figures C.19 – C.36 show the similar plots using measurements \( Y_2^{tr}, \ldots, Y_{10}^{tr} \).

The behavior of sign-change near \( h_0 \) is more distinctly and consistently visible through these figures. The curves based on the functional norm, \( Y(h, \lambda_0) \), seems to suggest that multiple sign-change points are possible.

### 2.4.5 Variability of the Diagnostic tools at \( \lambda = \lambda_0 \)

Through the Sections 2.2.3 and 2.2.4 we have concentrated on the behaviors of the GCV and functional norm individually. Then, we developed the two diagnostic tools
Figure 2.9: Diagnostic tool based on GCV \((2.4.1)\): \(\Upsilon(h, \lambda_0)\) as a function of \(\log(h)\) using \(\mathbf{Y}_t^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines at \(\log(h_0)\) (vertical) and 0 (horizontal).

Figure 2.10: Diagnostic tool based on functional norm \((2.4.2)\): \(\Upsilon(h, \lambda_0)\) as a function of \(\log(h)\) using \(\mathbf{Y}_t^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines at \(\log(h_0)\) (vertical) and 0 (horizontal).
and explored them in Sections 2.4.1–2.4.4. In all these, we looked at the behaviors based on single realizations of the process. Of course, we also compared them across ten realizations through the various Figures in the Appendices. But, as mentioned earlier in Section 2.2.2, we had in-fact generated 1000 realizations of the process and in this Section we examine the distributions of the two tools, $\Gamma$ and $\Upsilon$ using these 1000 realization.

As earlier, we restrict to the case $\lambda = \lambda_0$ and at each model bandwidth, $h$, we use boxplots to represent the distributions. Then we compare these boxplots as a function of $\log(h)$. Figure 2.11 shows the distribution of the tool based on GCV, $\Gamma(h, \lambda_0)$, as a function of $\log(h)$ and Figure 2.12, similarly, shows the distribution for the tool based on functional norm, $\Upsilon(h, \lambda_0)$. In both these figures, the first column of plots show a large range of $h \in [1.5, 60]$ and the second column of plots shows a smaller range near $h_0$, $h \in [h_0/2, 2h_0]$. The three rows in each figure correspond to the three values of the “true” bandwidth: $h_0 = 3$ (top row), $h_0 = 14.6$ (middle row), and $h_0 = 26.1$ (bottom row). As earlier, $\lambda_0 = 0.01$ is kept fixed.

Considering Figure 2.11, we observe that the distribution of the difference, $\Gamma(h, \lambda_0)$, is centered near 0 for $h$ near the true bandwidth, $h_0$. And as we move away from $h_0$ the distribution of the difference shifts to the positive side for $h > h_0$ and to the negative side for $h < h_0$, although there is considerable variability for $h \ll h_0$. Further, near the true bandwidth $h_0$, the distribution seems to have the smallest variability. In small neighborhoods around $h_0$, there is not much change in the spread of these distributions, yet the the location shifts very characteristically to either side.

The distributions of $\Upsilon(h, \lambda_0)$, as shown in Figure 2.12, is not as appealing as that of $\Gamma(h, \lambda_0)$. Although we see a similar shift of the center of the distribution in small
neighborhoods around $h_0$ as we move away considerably away from $h_0$, the behavior seems to reverse. For example, for $h \ll h_0$, (as seen from the plot for $h_0 = 26.1$) the distribution of the difference seems to start shift up toward zero and similarly for $h \gg h_0$ (as seen from the plot for $h_0 = 3$) the distribution seems to shift down toward zero. Further, the distribution with the smallest spread seems to observed consistently at a bandwidth smaller than $h_0$.

2.5 Concluding Remarks on the Simulation Experiments

In this chapter we have experimentally explored the typical regression framework using the mechanism of classical regularization and Gaussian Process mechanisms (Section 2.1) and utilized concepts (Section 2.2) from both to construct diagnostic tools (Section 2.4) which provide some information about the underlying data generation process.

We considered various underlying generation processes (using multiple $h_0$ values) for the purpose. We fit a myriad of training models to the generated data to understand the two primary concepts of GCV and functional norm which eventually led to the construction the diagnostic tools, $\Gamma$ and $\Upsilon$. Throughout the explorations, we have used figures based on two ranges of the model bandwidth values, $h$; the larger range providing a “big-picture” overview and the smaller range zooming in near the true bandwidth. These helped us visualize the various patterns clearly. To facilitate in the understanding further, we first explored the patterns on a $h \times \lambda$ grid and then restricted to the marginal at $\lambda = \lambda_0$. The Figures in the main text were obtained using a single realization of the generation process, $Y_1^{tr}$. But, to help us understand the consistency of the observed patterns all the experiments were repeated on multiple
Figure 2.11: Variability in the diagnostic tool based on GCV, $\Gamma(h, \lambda_0)$, as a function of $\log(h)$. Variability is depicted using 1000 repetitions, $Y_{1}^{tr}, \ldots, Y_{1000}^{tr}$. First column: $1.5 \leq h \leq 60$; second column: $h_0/2 \leq h \leq 2h_0$. Rows: $h_0 = 3$ (top), $h_0 = 14.6$ (middle), and $h_0 = 26.1$ (bottom). Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
Figure 2.12: Variability in the diagnostic tool based on functional norm, $\mathcal{Y}(h, \lambda_0)$, as a function of $\log(h)$. Variability is depicted using 1000 repetitions, $\mathcal{Y}_1^\text{tr}, \ldots, \mathcal{Y}_{1000}^\text{tr}$. First column: $1.5 \leq h \leq 60$; second column: $h_0/2 \leq h \leq h_0$. Rows: $h_0 = 3$ (top), $h_0 = 14.6$ (middle), and $h_0 = 26.1$ (bottom). Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
multiple realizations, \( Y_2^{tr}, \ldots, Y_{10}^{tr} \), the figures for which are provided in the Appendices. In fact, to understand the variability of the two tools developed we considered 1000 independent realizations and explored their distributions (Section 2.4.5).

We observed that the sign of the diagnostic tools is crucial from the point of view of “learning the kernel” problem. Especially, the tool based on GCV, \( \Gamma(h, \lambda_0) \), had very characteristic distribution depending on which side of the “true” bandwidth, \( h_0 \), was being explored.

With these observations, we examine the theoretical properties of these two diagnostic tools in the next chapter. Since, the strongest of the patterns was observed for the center of the distributions, we focus on the theoretical expectations first in this thesis.
Chapter 3: Understanding the Theoretical Behavior of the Diagnostic Tools

In this Chapter we discuss some of the theoretical aspects of the two diagnostic tools developed in the previous Chapter to provide some justification for the patterns observed earlier. The first section reviews some theory and results from the existing literature needed for this purpose. Most importantly, the concept of “majorization” relationship between two real vectors and the associated results with the Hadamard (Schür) product of matrices, are discussed. In the following section we present some simple consequences of the existing results in a terminology that is more in-tune to our work. In the third section we compute the expected values of the two diagnostic tools developed earlier and discuss their behavior on the $h \times \lambda$ grid and at $\lambda_0$ along the similar path as we have done in the previous Chapter. We briefly touch upon the aspect of varying the domain points and the “true” noise variances ($\lambda_0$) towards the end of this section. In Section 3.4, we move away from the parametric representation of the kernel functions to a more non-parametric flavor and discuss the main theoretical results. Finally, in the last section we summarize the work and results of this Chapter.
3.1 Introduction and Background

This section introduces the Schür (Hadamard) product of two matrices and discusses some results from literature pertinent to “learning the kernel” problem. Consider two matrices, $A = [a_{ij}]$ and $B = [b_{ij}]$, their Hadamard product is defined as $A \circ B := [a_{ij}b_{ij}]$. It is known that if $A$ and $B$ are positive definite Gram matrices, then their Hadamard product, $A \circ B$, is also a positive definite Gram matrix (since the product of two Mercer’s kernels is a Mercer’s kernel, see, e.g., Shawe-Taylor and Cristianini (2004, Prop. 3.2.2(iii), p.75)). In fact, the main reason to look at Hadamard products is that the Gram matrices corresponding to the product of two kernels is the Hadamard product of the individual Gram matrices. First we look at families of kernels indexed by one parameter that are closed under the Hadamard-product operation. Then, we provide a brief survey of relationships among the eigenvalues of $A, B$ and $A \circ B$.

3.1.1 Kernels closed under Hadamard product

Consider a generic family of kernels, $\mathcal{K}_\Theta := \{\kappa_\theta(\cdot, \cdot) : \theta \in \Theta\}$, indexed by $\theta$, for some parametric set $\Theta$. $\mathcal{K}_\Theta$ is said to be “closed under Hadamard product” if, for any $\theta_1, \theta_2 \in \Theta$, we get, $\kappa_{\theta_1} \cdot \kappa_{\theta_2} \in \mathcal{K}_\Theta$. Below, we discuss a few examples, some of which are “closed” under this sense and some are not:

1. **Polynomial kernels.** For a fixed $c \in \mathbb{R}^+$, let $\Theta_c = \mathbb{N}$ and $\kappa_\theta(x, y) := (\langle x, y \rangle + c)^\theta$. In this example, $c$ is held fixed and only the degree of the polynomial changes. Thus, $\kappa_{\theta_1} \cdot \kappa_{\theta_2} = (\langle x, y \rangle + c)^{\theta_1 + \theta_2}$ clearly belongs to $\mathcal{K}_\Theta$. 
2. Complete polynomial kernels. Let $\Theta = \mathbb{R}^+ \times \mathbb{N}$, $\theta = (c,p) \in \Theta$, and $\kappa_{(c,p)}(x,y) := (\langle x,y \rangle + c)^p$. Here, the constant term is also in the parametric space. In this case, for $c_1 \neq c_2$, $\kappa_{c_1,p_1} \cdot \kappa_{c_2,p_2} \notin \mathcal{K}_\Theta$. This follows from the fact that over the field of reals, factorization of polynomials into irreducible monics is unique (see for example, [Artin 1991] Thm. 1.5, p. 390).

3. RBF kernels. Let $\Theta = \mathbb{R}^+$ and $\kappa_{\theta}(x,y) := \exp\{-\theta^{-1}\|x-y\|^2\}$. Again, trivially, $\kappa_{\theta_1} \cdot \kappa_{\theta_2} = \exp\{-\theta_1^{-1} + \theta_2^{-1}\|x-y\|^2\}$ belongs to $\mathcal{K}_\Theta$.

4. Mahalanobis kernel. Let $\Theta = \mathbb{R}^{+d}$ and $\kappa_{\theta}(x,y) := \exp\{-(x-y)'\Lambda(x-y)\}$, where $\Lambda = \text{diag}(\theta_1, \ldots, \theta_d)$, $\theta_i \in \mathbb{R}^+ \forall i$. Although, strictly speaking, this is a multi-parameter family, it is easy to see that, $\kappa_{\theta_1} \cdot \kappa_{\theta_2} = \exp\{-(\theta_1^{-1} + \theta_2^{-1})\|x-y\|^2\}$ clearly belongs to $\mathcal{K}_\Theta$, where, $\Lambda_1 + \Lambda_2 = \text{diag}(\theta_1^{(1)} + \theta_2^{(1)}, \ldots, \theta_1^{(d)} + \theta_2^{(d)})$.

5. Fisher kernels These are defined as, $\kappa_{\theta}(x,y) := s_\theta(x)^I_\theta^{-1}s_\theta(y)$ where $s_\theta(x)$ is the score vector and $I_\theta$ is the information matrix corresponding to the parameter vector $\theta$. Even for univariate $\theta$ and exponential families Fisher’s kernel is not closed under Hadamard product. To see this, and to further simplify the situation, ignore $I_\theta$, and consider the naive Fisher kernel, $\kappa_{\theta}(x,y) := s_\theta(x)s_\theta(y)$. Further consider the natural exponential family, given by, $f(x; \theta) = c(x) \exp\{\theta x - A(\theta)\}$. Then $s_{\theta}(x) = (x - A'(\theta))$. For the closure property to hold we “essentially need,”

$$(x - A'(\theta_1))(x - A'(\theta_2)) = (x - A'(\theta_3)) \ \forall x, \ \forall \theta_1, \theta_2,$$

where $\theta_3$ (depends on $\theta_1, \theta_2$) belongs the parametric set defining the natural exponential family. Clearly this cannot happen.
6. **Kernels generated from kernels:** Two generic way to construct kernels which are closed under the Hadamard product are:

(a) Given any Mercer’s kernel $\kappa(\cdot, \cdot)$, let, $\Theta = \mathbb{N}$ and $K_{\Theta,k} := \{k^p(\cdot, \cdot) : p \in \Theta\}$. It is easy to check that $k^p(\cdot, \cdot)$ is a Mercer’s kernel (Herbrich 2001, Thm. 2.20). This encompasses the polynomial kernels discussed above.

(b) Consider a set $\Theta$ and an associated parametric class of continuous functions, $F_\Theta := \{f_\theta : \mathcal{X} \to \mathbb{R} : \theta \in \Theta\}$ closed under products, that is, $f_{\theta_1} f_{\theta_2} \in F_\Theta \forall \theta_1, \theta_2 \in \Theta$. Then, let $K_\Theta := \{\kappa_\theta(x,y) := f_\theta(x)f_\theta(y) : \theta \in \Theta\}$. In this case, it is easy to check that $\kappa_\theta$ is a Mercer’s kernel (Herbrich 2001, Thm. 2.20) and additionally, $K_\Theta$ is closed under the Schür product (by the virtue of $F_\Theta$). Although, in general, the polynomials and zero-mean Gaussian functions enjoy such “closure property,” the polynomial and RBF kernels, as defined above, are not generated by this type of construction.

In passing, it should be noted that the “kernels” used in the density estimation literature need not necessarily be positive definite and hence such kernels are not useful in the context of “kernel machines,” or “RKHS.” Two such examples are (Sriperumbudur et al. 2008):

- **Uniform (naive) kernels,** defined by $\kappa_\theta(x) = I(\|x\|^2 \leq \theta)$ for $\theta \in \mathbb{R}^+$ and

- **Epanechnikov kernels,** defined by $\kappa_\theta(x) = (\theta - \|x\|^2)I(\|x\|^2 \leq \theta)$ for $\theta \in \mathbb{R}^+$. 

64
3.1.2 Relationships between eigenvalues of $A$, $B$, and $A \circ B$ through the lens of Majorization

In this section we gather results and tools from the literature, especially, the various inequalities relating the spectra of two matrices. Some of these are used in this thesis and the other we hope to be useful in the future work.

The Hadamard product is associative and distributes over matrix addition. The “identity” element with respect to Hadamard-multiplication is the matrix $J = 11'$. Additionally, since the underlying ring in all our discussions, $\mathbb{R}$ or $\mathbb{C}$, is commutative, the Hadamard product is also commutative. Generally in the literature, complex matrices are considered, but for the current context, we will restrict to real symmetric matrices, and whenever appropriate, positive definite conditions will be assumed.

Let $M_n$ denote the class of all $n \times n$ real symmetric matrices. For $A \in M_n$, let $\lambda(A) = (\lambda_1(A), \ldots, \lambda_n(A))'$ denote the eigenvalues of $A$, where, $\lambda_{\max}(A) = \lambda_1(A) \geq \ldots \lambda_n(A) = \lambda_{\min}(A)$. The spectral radius of $A \in M_n$ is defined as,

$$\rho(A) := \max\{ |\lambda| : \lambda \text{ is an eigenvalue of } A \}.$$ 

For positive semi-definite matrices,

$$\rho(A) = \lambda_{\max}(A).$$

The spectral norm is defined as,

$$|||A||| := \max\{||Ax|| : x \in \mathbb{R}^n \text{ and } ||x|| = 1\};$$

since $A$ is symmetric,

$$|||A||| = \max\{|\lambda_{\min}(A)|, |\lambda_{\max}(A)|\};$$
and for positive semi-definite $A$,

$$||A|| = \lambda_{\text{max}}(A) = \rho(A).$$

The following is an adaptation of the Shcur’s Theorems (Horn 1990, Thm 3.1) in the current context.

**Theorem 8** (Schur’s Theorem). Let $A, B \in M_n$ be two positive semi-definite matrices, then.

(a) $A \circ B$ is positive semi-definite.

(b) $(\min a_{ii}) \lambda_{\text{min}}(B) \leq \lambda_{\text{min}}(A \circ B) \leq (\max a_{ii}) \lambda_{\text{max}}(B)$.  

(c) If $B$ is positive definite, $a_{ii} > 0 \forall i$, then $A \circ B$ is positive definite.

(d) $\lambda_{\text{max}}(A \circ B) = ||A \circ B|| \leq ||A|| \cdot ||B|| = \lambda_{\text{max}}(A) \lambda_{\text{max}}(B)$

Consider the following four point set:

$E(A, B) := \{\lambda_{\text{min}}(A)\lambda_{\text{min}}(B), \lambda_{\text{min}}(A)\lambda_{\text{max}}(B), \lambda_{\text{max}}(A)\lambda_{\text{min}}(B), \lambda_{\text{max}}(A)\lambda_{\text{max}}(B)\}$.  

With only symmetry assumption on $A$ and $B$, the following holds (Horn 1990, p. 90):

$$\min \{\lambda \in E(A, B)\} \leq \lambda_{\text{min}}(A \circ B) \leq \lambda_{\text{max}}(A \circ B) \leq \max \{\lambda \in E(A, B)\},$$

and with the additional positive semi-definite assumption, the following is due to Schür (see, e.g., Marshall et al. 2009, J.1. Theorem, p. 353)

$$\lambda_{\text{min}}(A)\lambda_{\text{min}}(B) \leq \lambda_{\text{min}}(A \circ B) \leq \lambda_{\text{max}}(A \circ B) \leq \max \{\lambda \in E(A, B)\}. $$
Marshall and Olkin (1979, H.1.a Theorem, p. 247) proved that for two positive semi-definite matrices, $A, B \in M_n$:

$$\prod_{i=k}^{n} \lambda_i(AB) \geq \prod_{i=k}^{n} \lambda_i(A)\lambda_i(B) \quad \forall k = 1, \ldots, n$$

and conjectured (Marshall and Olkin 1979, p. 258, Eqn. 15) that

$$\prod_{i=k}^{n} \lambda_i(A \circ B) \geq \prod_{i=k}^{n} \lambda_i(A)\lambda_i(B) \quad \forall k = 1, \ldots, n,$$


$$\prod_{i=k}^{n} \lambda_i(A \circ B) \geq \prod_{i=k}^{n} \lambda_i(AB) \quad \forall k = 1, \ldots, n.$$

Choosing, $k = 1$, gives the Oppenheim’s inequality,

$$\prod_{i=1}^{n} \lambda_i(A \circ B) = \det(A \circ B) \geq (\det A)(\det B) = \det(AB) = \prod_{i=1}^{n} \lambda_i(AB)$$

and Horn (1990, 3.3.9b) had proved the inequality for $k = n$:

$$\lambda_{\min}(A \circ B) \geq \lambda_{\min}(AB).$$

Cheng et al. (2007), determine the conditions on $A$ and $B$ such that

$$\prod_{i=1}^{k} \lambda_{n-i+1}(AB) = \prod_{i=1}^{k} (A \circ B) \quad \forall k = 1, \ldots, n.$$

**Theorem 9** (Hadamard Factorization Theorem). A given positive semi-definite matrix $C \in M_n$ can be factored as $C = A \circ B$ where $A$ is a positive semi-definite matrix and $B$ positive definite if and only if the number of positive main diagonal entries of $C$ is equal to the rank of $C$. 

67
In the context of gram matrices, which are positive definite, we can say that every positive definite matrix \( C \in M_n \) has a Hadamard factorization \( C = A \circ B \) with both \( A \) and \( B \) positive definite. Additionally, if \( C \) is symmetric then both \( A \) and \( B \) can be chosen to be symmetric as well.

The following lemma which provides bounds for the trace of product of two symmetric matrices is a due to von-Neumann Theorem:

**Lemma 1.** (see, e.g., [Rao and Rao 1998, P11.4.5, p. 386]) With only symmetry assumption on \( A \) and \( B \), the following holds

\[
\sum_{i=1}^{n} \lambda_i(A)\lambda_{n-i+1}(B) \leq \text{tr}(AB) \leq \sum_{i=1}^{n} \lambda_i(A)\lambda_i(B).
\]

To further understand the behavior of eigenvalues of these matrices, the concept of Majorization is needed:

**Definition 4** (Majorization). Consider two vectors,

\[
a = (a_1, \ldots, a_n) \quad \text{and} \quad b = (b_1, \ldots, b_n)
\]

in \( \mathbb{R}^n \). Let \( a_{[1]} \geq \ldots \geq a_{[n]} \) be the ordered coordinates of \( a \). Similarly for \( b \). Then \( a \) is said to be majorized by \( b \), denoted as, \( a \prec b \), if

\[
\sum_{i=1}^{k} a_{[i]} \leq \sum_{i=1}^{k} b_{[i]}, \quad k = 1, \ldots, n - 1, \quad \text{and}
\]

\[
\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i
\]

If the last equality is an inequality instead, that is, \( \sum_{i=1}^{n} a_i \leq \sum_{i=1}^{n} b_i \), then \( a \) is said to be weakly majorized by \( b \), denoted as, \( a \prec_w b \).

The following Lemma was proved by Hardy, Littlewood, and Pólya and can be found in [Marshall et al. (2009), A.3. Proposition, p. 207].


Lemma 2. Consider two vectors, $a, b \in \mathbb{R}^n$. Let $a[1] \geq \ldots \geq a[n]$ be the ordered coordinates. Similarly for $b$. Then,

$$
\sum_{i=1}^{n} a[i] b[n-i+1] \leq \sum_{i=1}^{n} a[i] b[i].
$$

(3.1.1)

The following Lemma is essentially the Birkhoff-von Neumann Theorem which provides a mechanism to express a majorized vector in terms of the majorizing vector and can be found in Marshall and Olkin (1979, C.1. Proposition, p. 113):

Lemma 3. Consider two vectors, $a, b \in \mathbb{R}^n$, where, $a_1 \geq \ldots \geq a_n$ and $b_1 \geq \ldots \geq b_n$. Then, $a \prec b \iff a$ lies in the convex hull of $n!$ permutations of $b$. In other words, if $\pi^l : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$, for $l = 1, \ldots, n!$, are all the permutations of $(1, \ldots, n)$ then,

$$
a \prec b \iff a = \sum_{l=1}^{n!} w_l b_{\pi^l},
$$

where, $w_l \geq 0$ and $\sum_l w_l = 1$ are the weights.

For two vectors on a simplex, one majorized by the other, the following Lemma gives an inequality between the inner products of these vectors with a third (ordered) vector. A proof for it can be found in Marshall et al. (2009, Prop. A.2.a, p639).

Lemma 4. Consider three vectors $a, b, x \in \mathbb{R}^n$ such that $x_1 \leq x_2 \leq \ldots \leq x_n$. Then,

$$
\sum_{i=1}^{n} a_i x_i \leq \sum_{i=1}^{n} b_i x_i
$$

(3.1.2)

if and only if

$$
\sum_{i=1}^{k} a_i \geq \sum_{i=1}^{k} b_i, \text{ for } k = 1, \ldots, n-1, \quad \text{and} \quad \sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i.
$$

(3.1.3)

The following Lemma provides an inequality for the coordinate wise evaluations of a convex function at two points, one majorized by the other. This is a special case
of what are called as the Schür-convex functions. A proof for this can be found in [Marshall and Olkin, 1979, Proposition C.1, p. 64].

**Lemma 5.** If \( g : \mathbb{I} \rightarrow \mathbb{R} \) is a convex function defined on an interval, \( \mathbb{I} \), and \( a \prec b \), then

\[
\sum_{i=1}^{n} g(a_i) \leq \sum_{i=1}^{n} g(b_i)
\]

The next lemma, due to Bapat and Sunder (1985, Corollary 2), is a corner stone in this research. It connects the eigenvalues of Hadamard products via majorization.

**Lemma 6.** Let \( A, B \in M_n \) such that \( A \) symmetric, \( B \) is p.s.d and \( b_{ii} = 1 \) for all \( i \). Then,

\[
\lambda(A \circ B) \prec \lambda(A).
\]

### 3.2 Some Simple Consequences

In this section we discuss at a few simple implications and corollaries of the results stated in the previous section. These statements are later used in Section 3.4 to prove the results therein.

Consider Lemma 1. If we assume \( A \) is positive definite, then we can replace \( A \) in the Lemma by an integer power of its inverse, \( A^{-p} \) for \( p \in \mathbb{N} \). Then, it follows trivially that

\[
\text{tr} (A^{-p}B) \geq \sum_{i=1}^{n} \frac{\lambda_i(B)}{(\lambda_i(A))^p}.
\]  

(3.2.1)

Now, consider Lemma 4. Assuming \( a_1 \geq a_2 \geq \ldots \geq a_n \) and \( b_1 \geq b_2 \geq \ldots \geq b_n \) makes the conditions in (3.1.3) equivalent to the condition \( b \prec a \). However, the Lemma itself does not require this assumption. Further, assume that \( a_n > 0 \), \( b_n > 0 \).
and let \( p \in \mathbb{N} \). Then, the following two inequalities follow from (3.1.2):

\[
\sum_{i=1}^{n} a_i^{-(p-1)} \leq \sum_{i=1}^{n} b_i a_i^{-p} \\
\sum_{i=1}^{n} a_i b_i^{-p} \leq \sum_{i=1}^{n} b_i^{-(p-1)}
\]

(3.2.2)

by choosing, \( x_i = b_i^{-p} \) and \( x_i = a_i^{-p} \) respectively.

The next Lemma is a direct application of Lemma 6 specifically for the Gram matrices.

**Lemma 7.** For a family of kernels closed under the Hadamard product operation if \( K_{\theta_1} = K_{\theta_2} \circ K_{\theta_*} \) for \( \theta_1, \theta_2, \) and \( \theta_* \in \Theta \), then,

\[
\lambda(K_{\theta_1}) \prec \lambda(K_{\theta_2}).
\]

In the case of two RBF kernels, as defined in the previous section, with bandwidths, \( \theta_1 < \theta_2 \), we can write, \( \frac{1}{\theta_1} = \frac{1}{\theta_2} + \frac{\theta_2 - \theta_1}{\theta_1 \theta_2} \). Thus, \( K_{\theta_1} = K_{\theta_2} \circ K_{\theta_*} \), where, \( K_{\theta_*} \) is the Gram matrix corresponding to the bandwidth \( \frac{\theta_1 \theta_2}{\theta_2 - \theta_1} \). Thus, \( \lambda(K_{\theta_1}) \prec \lambda(K_{\theta_2}) \).

Further, this result can be extended to convex combination of kernels, as stated below.

**Result 2.** Consider three RBF kernels with bandwidths \( \theta_1 < \theta_2 < \theta_3 \). Then, for \( 0 < \alpha < 1 \),

\[
\lambda(\alpha K_{\theta_1} + (1 - \alpha) K_{\theta_2}) \prec \lambda(K_{\theta_3}).
\]

(3.2.3)

(Note that, \( \alpha \kappa_{\theta_1}(\cdot, \cdot) + (1 - \alpha) \kappa_{\theta_1}(\cdot, \cdot) \) is not a RBF kernel anymore.)

**Proof.** Since, \( \theta_1 < \theta_3 \), \( K_{\theta_1} = K_{\theta_3} \circ K_{\theta_{31}} \). Similarly, \( K_{\theta_2} = K_{\theta_3} \circ K_{\theta_{32}} \). Here, \( \theta_{3i} > 0 \) for \( i = 1, 2 \). It is easy to check that,

\[
\alpha K_{\theta_1} + (1 - \alpha) K_{\theta_2} = K_{\theta_3} \circ K_{\theta_{3*,}}
\]
where, $K_{\theta_3} = \alpha K_{\theta_{31}} + (1 - \alpha) K_{\theta_{32}}$. Clearly, $K_{\theta_3}$ is positive-semi-definite with diagonal entries all one. Thus, invoking Bapat and Sunder (1985, Corollary 2), Lemma 6 above, we have,

$$\lambda(\alpha K_{\theta_3} + (1 - \alpha) K_{\theta_2}) = \lambda(K_{\theta_3} \circ K_{\theta_3}) \prec \lambda(K_{\theta_3}).$$

Clearly, the stated result is not very specific to RBF kernels and one can easily generalize (3.2.3) to non-RBF kernels, as long as existence of $\theta_{31}, \theta_{32} \in \Theta$ are guaranteed.

Further, it is worthwhile to note that the above result might be useful to the researchers utilizing the semi-definite programming techniques, as discussed in Section 1.6.1 to search for “optimal” kernels from a convex hull of pre-specified kernels.

### 3.3 Moments of the Diagnostic Tools

Note that the two diagnostic tools developed in Chapter 2, \(\Gamma(2.4.1)\) and \(\Upsilon(2.4.2)\), are in fact quadratic forms in the the measurements, \(Y^{tr}\). In general, for a quadratic form, \(Y^{t}MY\), assuming \(EY = 0\), the first moment is given by \(E(Y^{t}MY) = tr\{MS\}\) where, \(V(Y) = \Sigma\). This, of course, only requires the existence of the first two moments of \(Y\). Additionally, if \(Y\) is Gaussian, we further have \(V(Y^{t}MY) = tr\{MSM\Sigma\}\), which in case of general stochastic processes involves fourth moments of \(Y\). Since, the first moment needs least distributional assumption, and moreover, motivated by the patterns observed from the simulation experiments of the previous chapter we first explore the theoretical expectations of the two diagnostic tools in this section.

In particular we want to understand, \(E_{h_0, \lambda_0} \Gamma(h, \lambda)\) and \(E_{h_0, \lambda_0} \Upsilon(h, \lambda)\).
Before we go into the details, in the next subsection we briefly discuss the behavior of the expectation of GCV and the functional norm under the modeling distribution and we get back to the expected values of the tools in the following subsection.

3.3.1 Monotonicity of GCV and Functional Norm Under the Model Distribution

In Section 2.3, we considered the expected values of GCV and functional norm under the modeling distribution as defined by (2.3.1) and (2.3.2). Let, \( \lambda(K_h) = (\gamma_1(h), \ldots, \gamma_n(h)) \) denote the eigenvalues of \( K_h \) where \( \gamma_1(h) \geq \gamma_2(h) \geq \cdots \geq \gamma_n(h) \). Then, we have the following two simple results.

**Result 3.** For \( \lambda > 0 \), \( E_{h,\lambda} GCV(h, \lambda) \) is a monotonically decreasing function of \( h \).

**Proof.** Consider the convex function,

\[
g^*(x) = \frac{1}{x + \lambda},
\]

for \( x \in [0, n] \) and define,

\[
g(h) = \sum_{i=1}^{n} g^*(\gamma_i(h)) = \sum_{i=1}^{n} \frac{1}{\gamma_i(h) + \lambda}.
\]

For \( h_1 < h_2 \), using Lemma 7 we have, \( \lambda(K_{h_1}) \prec \lambda(K_{h_2}) \). Since, \( g^*(\cdot) \) is convex, using Lemma 5 we get,

\[
g(h_1) < g(h_2).
\]

Hence,

\[
E_{h_1,\lambda} GCV(h_1, \lambda) = \frac{1}{g(h_1)} > \frac{1}{g(h_2)} = E_{h_2,\lambda} GCV(h_2, \lambda).
\]

**Result 4.** For \( \lambda > 0 \), \( E_{h,\lambda} \nu(h, \lambda) \) is a monotonically decreasing function of \( h \).
Proof. Consider the convex function,

\[ g^*(x) = -\frac{x}{x + \lambda}, \]

for \( x \in [0, n] \) and define,

\[ g(h) = \sum_{i=1}^{n} g^*(\gamma_i(h)) = -\sum_{i=1}^{n} \frac{\gamma_i(h)}{\gamma_i(h) + \lambda}. \]

Then, following a similar line of argument as the previous result, we get the desired result:

\[ E_{h_1, \lambda} \nu(h_1, \lambda) = -g(h_1) > -g(h_2) = E_{h_2, \lambda} \nu(h_2, \lambda). \]

3.3.2 Expected Values of the Diagnostic Tools

Recall from (2.4.1) and (2.4.2) that

\[ \Gamma(h, \lambda) = \text{tr} \left\{ B_h,\lambda \left( (K_h + \lambda I)^{-1} D_{Ytr} - I \right) \right\}, \]

\[ \Upsilon(h, \lambda) = \text{tr} \left\{ A_h,\lambda \left( (K_h + \lambda I)^{-1} D_{Ytr} - I \right) \right\}. \]

Thus, under the true distribution, we have,

\[ E_{h_0, \lambda_0} \Gamma(h, \lambda) = \text{tr} \left\{ B_{h_0,\lambda_0} \left( (K_{h_0} + \lambda_0 I)^{-1} (K_{h_0} + \lambda_0 I) - I \right) \right\}, \]

\[ E_{h_0, \lambda_0} \Upsilon(h, \lambda) = \text{tr} \left\{ A_{h,\lambda_0} \left( (K_{h_0} + \lambda_0 I)^{-1} (K_{h_0} + \lambda_0 I) - I \right) \right\}. \]

At \( \lambda = \lambda_0 \), we get,

\[ E_{h_0, \lambda_0} \Gamma(h, \lambda_0) = \text{tr} \left\{ B_{h,\lambda_0} \left( (K_{h_0} + \lambda_0 I)^{-1} (K_{h_0} + \lambda_0 I) - I \right) \right\}, \]

\[ E_{h_0, \lambda_0} \Upsilon(h, \lambda_0) = \text{tr} \left\{ A_{h,\lambda_0} \left( (K_{h_0} + \lambda_0 I)^{-1} (K_{h_0} + \lambda_0 I) - I \right) \right\}. \]

To better understand the patterns observed earlier, we explore these expected values as a function of \( h \) and \( \lambda \). In particular, we are interested in their sign-change.
behavior, especially at $\lambda_0$. In this respect, \([3.3.1]\) is equivalent to

$$E_{h_0,\lambda_0} \Gamma^*(h, \lambda_0) = \text{tr} \left\{ (K_h + \lambda_0 I)^{-1} \left( (K_h + \lambda_0 I)^{-1} (K_{h_0} + \lambda_0 I) - I \right) \right\},$$

ignoring the positive factor $n (\text{tr} \{(K_h + \lambda_0 I)^{-1}\})^{-2}$. Further, note that, the expectation of the “discrepancy-decorrelation” matrix, $\Xi$, defined in \([2.4.3]\),

$$E_{h_0,\lambda_0} \Xi(h, \lambda) = (K_h + \lambda I)^{-1}(K_{h_0} + \lambda_0 I) - I$$

is common to all these expressions.

### 3.3.3 Behavior of the Expectations at $\lambda = \lambda_0$

![Graphs showing behavior of the Expectations at $\lambda = \lambda_0$](image)

Figure 3.1: Expected diagnostic tool based on GCV \([3.3.1]\): $E_{h_0,\lambda_0} \Gamma(h, \lambda_0)$ as a function of $\log(h)$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
Figure 3.2: Expected diagnostic tool based on functional norm (3.3.2): $E_{h_0,\lambda_0} \gamma(h, \lambda_0)$ as a function of $\log(h)$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

Figure 3.1 shows the expected values of the diagnostic tool based on GCV, given by (3.3.1), as a function of $\log(h)$. The objective is to study the sign-change behavior of these curves. The horizontal gray line is at 0. We observe that, for all the different values of $h_0$ considered (including the ones not shown), the sign of $E_{h_0,\lambda_0} \gamma(h, \lambda_0)$ changes at $h_0$. Similar behavior is also observed for the expected value of the diagnostic tool based on functional norm, $E_{h_0,\lambda_0} \gamma(h, \lambda_0)$, as given by (3.3.2), and shown in Figure 3.2. Although, on a closer observation, $E_{h_0,\lambda_0} \gamma(h, \lambda_0)$ seems to display multiple sign-change possibilities. Further, inspecting the bottom row plots in both these figures, we see that the differences are monotonic in small neighborhood around $h_0$. 
3.3.4 Behavior of the Expectations on $h \times \lambda$ Grid

Now, to evaluate the behavior of these tools at $\lambda \neq \lambda_0$ consider, Figures 3.3 and 3.4 which shows the differences $E_{h_0,\lambda_0} \Gamma(h, \lambda)$ and $E_{h_0,\lambda_0} \Upsilon(h, \lambda)$ respectively. As earlier, the solid curve in these plots, approximately, denotes the boundary where the corresponding differences change sign and the vertical and horizontal gray lines are at $\log(h_0)$ and $\log(\lambda_0)$, respectively. A pattern that is clearly and consistently visible is that at $\lambda < \lambda_0$ the sign change occurs at a $h^* < h_0$ and for $\lambda > \lambda_0$ the sign-change occurs at a $h^* > h_0$. This is similar to the behavior observed in the previous chapter and is not surprising. Using a regularization coefficient $\lambda < \lambda_0$, imposes small penalty on the norm of the estimated function. Thus, the training over-fits the data. In other words, the estimated function is less smooth than the “true” underlying generator. Similarly, using $\lambda > \lambda_0$ imposes larger penalty on the norm which results in over-smoothing, that is the estimated function is more smoother than the generator. In both the cases, the estimated function has bad generalization errors which is visible from large GCV and MSEv values (e.g., see Figures 2.1, 2.3, A.20, and A.30).

3.3.5 Varying Training Sets and Noise Variances

Up to this point we have looked the diagnostic tools for a fixed training set and a fixed noise variance, $\lambda_0$. In what follows, we very briefly explore the behavior of the expected values of these tools at completely independent replications of the training set and a few different values of the true noise variance $\lambda_0$. That is, we are interested in the sign changing behavior of $E_{h_0,\lambda_0} \Gamma(h, \lambda_0)$, given by (3.3.1), and $E_{h_0,\lambda_0} \Upsilon(h, \lambda_0)$ given by (3.3.2), as a function of $h$ by changing training sets and noise variances.
Figure 3.3: Expected diagnostic tool based on GCV \([3.3.1]\): \(E_{h_0,\lambda_0} \Gamma(h, \lambda)\) on \(h \times \lambda\) grid. Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines at \(\log(h_0)\) (vertical) and \(\log(\lambda_0)\) (horizontal).

Figure 3.4: Expected diagnostic tool based on functional norm \([3.3.2]\): \(E_{h_0,\lambda_0} \Upsilon(h, \lambda)\) on \(h \times \lambda\) grid. Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines at \(\log(h_0)\) (vertical) and \(\log(\lambda_0)\) (horizontal).
Figure 3.5: Expected diagnostic tool based on GCV (3.31): \( E_{n, \lambda_0}(T(h, \lambda_0)) \) as a function of \( \log(h/h_0) \). Here \( h = 39.69 \) was kept fixed. The subplots in each row correspond to different noise levels, specified at the top. There are 10 different curves in each subplot corresponding to independent sets of training points. The bottom row zooms in on the top row such that \(-1 \leq \log(h/h_0) \leq 1\).
Consider the simulation setup described in Section 2.2.2. For the experiments in this section we use six noise variances, $\lambda_0 \in \{0.01, 0.09, 0.49, 1, 4, 9\}$, fifty point for the true bandwidth $\log(h_0) \in [-2, 2]$, and unlike the earlier experiments, the model bandwidth is fixed at, $h = 39.69$, which was the median of the squared Euclidean distances between the domain points. Note that, $\lambda_0$ is the signal to noise ratio, so statistically, it makes sense to only consider $\lambda_0 < 1$. But since we are also interested in exploring $E_{h_0, \lambda_0} \Gamma(h, \lambda_0)$ and $E_{h_0, \lambda_0} \Upsilon(h, \lambda_0)$ as algebraic objects, we choose to investigate $\lambda_0 > 1$ as well. The expected values of the two tools is plotted as a function of $\log(h/h_0)$. Figure 3.5 shows $E_{h_0, \lambda_0} \Gamma(h, \lambda_0)$ while Figure 3.6 shows $E_{h_0, \lambda_0} \Upsilon(h, \lambda_0)$. The process was repeated ten times with independent sets of training points, corresponding to the ten curves within each subplot. The ten curves within each subplot are not distinguishable due to the large values of $h$ and $h_0$ considered. At lower values of $h$ and $h_0$, these would be visibly distinguishable.

Clearly, similar to Figures 3.1 and 3.2 when $h_0 > h$ (that is, $\log(h/h_0) < 0$), both $E_{h_0, \lambda_0} \Gamma(h, \lambda_0) < 0$ and $E_{h_0, \lambda_0} \Upsilon(h, \lambda_0) < 0$ for all the repetitions considered. As $h_0$ comes closer to $h$, the difference gets closer to zero and for $h_0 \leq h$ (that is, $\log(h/h_0) \geq 0$), $E_{h_0, \lambda_0} \Gamma(h, \lambda_0) \geq 0$. In the case of $E_{h_0, \lambda_0} \Upsilon(h, \lambda_0)$, we observe that for large values of $\lambda_0$, that is, for extremely noisy measurements, there are multiple sign-change points. In fact, if we use even larger values of $\lambda_0$ (such as $\lambda_0 = 30$), we observe a reversal of the signs with respect to $h$, that is, $E_{h_0, \lambda_0} \Upsilon(h, \lambda_0) > 0$ for $h < h_0$ and $E_{h_0, \lambda_0} \Upsilon(h, \lambda_0) > 0$ for $h < h_0$; which is contrary to what we have observed all through the previous experiments.

The entire experiment was repeated for a few different values of $d$ (using appropriate number of training points, $n$); each yielding similar results. The value of $\lambda_0$ at
which multiple sign changes were observed for $E_{h_0, \lambda_0} \mathcal{T}(h, \lambda_0)$ depended on the inter-point distances of the domain points. For example, with $d = 1$, and Uniformly generated domain points, $\lambda_0 = 0.49$, was large enough for sign reversals of $E_{h_0, \lambda_0} \mathcal{T}(h, \lambda_0)$. Yet, in all the cases the pattern of $E_{h_0, \lambda_0} \Gamma(h, \lambda_0)$, remained unchanged.

### 3.4 Going Beyond Parametric Kernels

All through the Chapter 2 and earlier sections we have considered a parametric representation of the kernels. But as we will shortly see, equipped with the majorization we can work with a non-parametric representation of the involved kernels. In this respect, let $\kappa_0$ and $\kappa_1$ denote the “true” and model kernels respectively with $K_0$ and $K_1$ being the corresponding Gram matrices (based on the same domain points). In this notation, we can rewrite some of the earlier expressions as follows.

For GCV,

$$\text{GCV}(\kappa_1, \lambda) := \text{tr} \left\{ B_{\kappa_1, \lambda} (K_1 + \lambda I)^{-1} \mathcal{D} \right\},$$

$$E_{\kappa_0, \lambda_0} \text{GCV}(\kappa_1, \lambda) = \text{tr} \left\{ B_{\kappa_1, \lambda} (K_1 + \lambda I)^{-1} (K_0 + \lambda_0 I) \right\},$$

$$E_{\kappa_1, \lambda} \text{GCV}(\kappa_1, \lambda) = \text{tr} \left\{ B_{\kappa_1, \lambda} (K_1 + \lambda I)^{-1} (K_1 + \lambda I) \right\} = \text{tr} \left\{ B_{\kappa_1, \lambda} \right\},$$

$$\Gamma(\kappa_1, \lambda) := \text{GCV}(\kappa_1, \lambda) - E_{\kappa_1, \lambda} \text{GCV}(\kappa_1, \lambda)$$

$$= \text{tr} \left\{ B_{\kappa_1, \lambda} \left( (K_1 + \lambda I)^{-1} \mathcal{D} - I \right) \right\},$$

$$E_{\kappa_0, \lambda_0} \Gamma(\kappa_1, \lambda) = \text{tr} \left\{ B_{\kappa_1, \lambda} \left( (K_1 + \lambda I)^{-1} (K_0 + \lambda_0 I) - I \right) \right\},$$

where,

$$B_{\kappa_1, \lambda} = \frac{n}{(\text{tr} \left\{ (K_1 + \lambda I)^{-1} \right\})^2} (K_1 + \lambda I)^{-1},$$
Figure 3.6: Expected diagnostic tool based on functional norm (3.3.2): $E_{h_0, \lambda_0} \mathcal{Y}(h, \lambda_0)$ as a function of $\log(h/h_0)$. Here $h = 1$ was kept fixed. The subplots in each row correspond to different noise levels, specified at the top. There are 10 different curves in each subplot corresponding to independent sets of training points. The bottom row zooms in on the top row such that $-1 \leq \log(h/h_0) \leq 1$. 
and for functional norm,
\[
\nu(\kappa_1, \lambda) := \text{tr} \left\{ A_{\kappa_1, \lambda} (K_1 + \lambda I)^{-1} D_{Y^{tr}} \right\},
\]
\[
E_{\kappa_0, \lambda_0} \nu (\kappa_1, \lambda) = \text{tr} \left\{ A_{\kappa_1, \lambda} (K_1 + \lambda I)^{-1} (K_0 + \lambda_0 I) \right\},
\]
\[
E_{\kappa_1, \lambda} \nu (\kappa_1, \lambda) = \text{tr} \left\{ A_{\kappa_1, \lambda} (K_1 + \lambda I)^{-1} (K_1 + \lambda I) \right\} = \text{tr} \left\{ A_{\kappa_1, \lambda} \right\};
\]
\[
\Upsilon(\kappa_1, \lambda) := \nu(\kappa_1, \lambda) - E_{\kappa_1, \lambda} \nu (\kappa_1, \lambda)
\]
\[
= \text{tr} \left\{ A_{\kappa_1, \lambda} \left( (K_1 + \lambda I)^{-1} D_{Y^{tr}} - I \right) \right\}
\]
\[
E_{\kappa_0, \lambda_0} \Upsilon(\kappa_1, \lambda) = \text{tr} \left\{ A_{\kappa_1, \lambda} \left( (K_1 + \lambda I)^{-1} (K_0 + \lambda_0 I) - I \right) \right\},
\]
where,
\[
A_{\kappa_1, \lambda} = (K_1 + \lambda I)^{-1} K_1.
\]

3.4.1 Revisiting classical regularization using the $\Xi(h, \lambda)$ matrix:

It is worthwhile to revisit the classical regularization framework from the perspective of the “decorrelation-discrepancy” matrix, $\Xi$, defined in (2.4.3). The parametric notation for the kernels and Gram matrices are not required for our purpose, so re-defining
\[
\Xi(\kappa_1, \lambda) := (K_1 + \lambda I)^{-1} D_{Y^{tr}} - I.
\]
we get,
\[
E_{\kappa_0, \lambda_0} \Xi(\kappa_1, \lambda) = (K_1 + \lambda I)^{-1} (K_0 + \lambda_0 I) - I.
\]
Note that the transformed measurements, $(K_0 + \lambda_0 I)^{-1/2} D_{Y^{tr}}$, are uncorrelated. But, we do not know $K_0$, so in the modeling space, $(K_1 + \lambda I)^{-1/2} D_{Y^{tr}}$, is our best attempt at decorrelating the measurements with $(K_1 + \lambda I)^{-1/2}(K_0 + \lambda_0 I)(K_1 + \lambda I)^{-1/2}$ as
its variance. Thus,

$$\text{tr}\{E_{\kappa_0,\lambda_0} \Xi(\kappa_1, \lambda)\} = \text{tr}\{(K_1 + \lambda I)^{-1/2}(K_0 + \lambda_0 I)(K_1 + \lambda I)^{-1/2}\} - n$$

is one way to measure the discrepancy between the decorrelation in model space and that in the “true” underlying space.

Note that in theory the RKHS regularization (the Representer Theorem) framework assumes that the true underlying space is same as the model space, $\kappa_1 = \kappa_0$, but the noise variance (or more precisely, the signal to noise ratio), $\lambda_0$, is unknown. We get,

$$E_{\kappa_0,\lambda} \Xi(\kappa_0, \lambda) = (K_0 + \lambda I)^{-1}(K_0 + \lambda_0 I) - I = (\lambda_0 - \lambda)(K_0 + \lambda_0 I)^{-1}.$$

Thus, the magnitude of $\text{tr}\{E_{\kappa_0,\lambda} \Xi(\kappa_0, \lambda)\}$ gives us a measure of how far the model $\lambda$ is from $\lambda_0$ and its sign indicates the direction. In this sense, as a data dependent tool, we can use

$$\text{tr}\{\Xi(\kappa_0, \lambda)\} = Y^{tr'}(K_0 + \lambda I)^{-1}Y^{tr} - n$$

to reduce the search space for $\lambda_0$ by weeding out the extreme values of $\lambda$. Then, within the reduced space, we can fall back to any of the cross-validation techniques to choose the “best” estimate of $\lambda_0$.

### 3.4.2 Theoretical Behavior of the Tool Based on GCV

The following Proposition provides a (one-sided) justification for the diagnostic tool based on GCV for evaluating “appropriateness” of the model bandwidth $h$ when the “true” generator distribution is based on an unknown $h_0$. In what follows, borrowing the notations from Section 3.1.2, $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$ denotes
the eigenvalues of a real symmetric matrix $A$. This is not to be confused with the penalization coefficient, $\lambda$, or the prior noise variance, $\lambda_0$.

Using the non-parametric representation of the kernels we prove the following:

**Proposition 1.** At $\lambda = \lambda_0$, for $\lambda(K_0) \prec \lambda(K_1)$, we have

$$E_{\kappa_0, \lambda_0} \Gamma(\kappa_1, \lambda_0) \geq 0.$$ 

**Proof.** Using Lemma 3,

$$\lambda_i(K_0) = \sum_{j=1}^{n!} w_j \lambda_{\pi'(j)}(K_1) \quad \text{for } i = 1, \ldots, n, \quad (3.4.2)$$

where, $\{\pi^j : j = 1, \ldots, n!\}$ are all the $n!$ permutations of $(1, \ldots, n)$. To simplify the notation, let $\gamma_{0i} = \lambda_i(K_0)$ and $\gamma_{1i} = \lambda_i(K_1)$ for $i = 1, \ldots, n$. Then,

$$\text{tr} \left\{ (K_1 + \lambda_0 I)^{-2} (K_0 + \lambda_0 I) \right\} \geq \sum_{i=1}^{n} \frac{\gamma_{0i} + \lambda_0}{(\gamma_{1i} + \lambda_0)^2} \quad \text{(Using (3.2.1))}$$

$$= \sum_{i=1}^{n} (\gamma_{1i} + \lambda_0)^{-2} \sum_{j=1}^{n!} w_j \left( \gamma_{1\pi'(j)} + \lambda_0 \right) \quad \text{(Using (3.4.2))}$$

$$= \sum_{j=1}^{n!} w_j \sum_{i=1}^{n} (\gamma_{1i} + \lambda_0)^{-2} (\gamma_{1\pi'(j)} + \lambda_0)$$

$$\geq \sum_{j=1}^{n!} w_j \sum_{i=1}^{n} (\gamma_{1i} + \lambda_0)^{-1} \quad \text{(Using (3.1.1))}$$

$$= \sum_{i=1}^{n} (\gamma_{1i} + \lambda_0)^{-1} = \text{tr} \left( K_1 \lambda_0 I \right)^{-1}.$$ 

Thus,

$$\text{tr} \left\{ B_{\kappa_1, \lambda_0} \left( (K_1 + \lambda_0 I)^{-1} (K_0 + \lambda_0 I) - I \right) \right\}$$

$$= c \text{tr} \left\{ (K_1 + \lambda_0 I)^{-2} \left( (K_0 + \lambda_0 I) - (K_1 + \lambda_0 I)^{-1} \right) \right\}$$

$$\geq 0,$$
where \( c > 0 \) is the scalar term in \( B_{\kappa_1, \lambda_0} \).

Note that, in general, for any two arbitrary kernels, \( \kappa_0 \) and \( \kappa_1 \), neither \( \lambda(K_0) \prec \lambda(K_1) \) nor \( \lambda(K_1) \prec \lambda(K_0) \) may hold. But for the product kernel, \( \kappa_0 \cdot \kappa_1 \), it follows from Lemma 6 that \( \lambda(K_0 \circ K_1) \prec \lambda(K_0) \) and \( \lambda(K_0 \circ K_1) \prec \lambda(K_1) \). In particular, as explained through Lemma 7, for a class of kernels parametrized by a single parameter and closed under the Hadamard product operation (e.g., RBF kernels), depending on the ordering between two parameters, one of the two majorization relationships, \( \lambda(K_0) \prec \lambda(K_1) \) or \( \lambda(K_1) \prec \lambda(K_0) \) will certainly hold. Further, as discussed in Results 5 and 6 below, one can construct special kernels to satisfy these properties as well.

Finally, in the case of Gaussian kernels, using Lemma 7, the above Proposition can be restated in a parametric form as

\[
h \geq h_0 \implies E_{h_0, \lambda_0} \Gamma(h, \lambda_0) \geq 0.
\]

To further explain the observed behavior of \( \Gamma \), we need to show that for \( \lambda(K_1) \prec \lambda(K_0) \) (that is, \( h < h_0 \) for Gaussian kernel), \( E_{\kappa_0, \lambda_0} \Gamma(\kappa_1, \lambda_0) \leq 0 \). We show that this holds for some special classes of kernels.

**Result 5.** For the equi-correlation kernel defined by

\[
\kappa_\rho(x_i, x_j) = \rho + (1 - \rho)\delta_{ij}, \quad \text{for} \ 0 \leq \rho \leq 1,
\]

where, \( \delta_{ij} \) is the Kronecker’s delta function, we have,

\[
\rho < \rho_0 \implies E_{\rho_0, \lambda_0} \Gamma(\rho, \lambda_0) < 0.
\]
Proof. The Gram matrix is given by $K_\rho = (1 - \rho)I + \rho J$. It is easy to check (see, e.g., [Rao 2002] Ex. 1.1, p. 67) that the eigenvalues for $K_\rho$ are $1 + (n - 1)\rho$ and $1 - \rho$ with multiplicities $1$ and $n - 1$, respectively. Clearly, for $0 < \rho_1 < \rho_2 < 1$,

$$K_{\rho_1} = K_{\rho_2} \circ K_{\rho_*},$$

where, $\rho_* = \rho_1/\rho_2 < 1$. Thus, $\kappa_{\rho_*}$ is a legitimate kernel in this family. Hence, using Lemma [7] we get $\lambda(K_{\rho_1}) \prec \lambda(K_{\rho_2})$. Even otherwise, it is easy to verify this majorization relationship between $\lambda(K_{\rho_1})$ and $\lambda(K_{\rho_2})$ directly from the following fact: for all $0 \leq j \leq n - 1$,

$$1 + (n - 1)\rho_1 + j(1 - \rho_1) \leq 1 + (n - 1)\rho_2 + j(1 - \rho_2),$$

with equality occurring only when $j = n - 1$.

Now, analogous to the setup in Proposition [1], let $\rho_0$ correspond to the “true” kernel and $\rho$ to the model kernel.

Let $U$ be the matrix of eigenvectors (as columns) for $K_\rho$. (The first column is $1/\sqrt{n}$.) It is known that $U$ is also the eigenvector matrix for $K_{\rho_0}$. Further, let $\Lambda_\rho = [\lambda_i(K_\rho)]_{n \times n} + \lambda_0 I$ and $\Lambda_{\rho_0} = [\lambda_i(K_{\rho_0})]_{n \times n} + \lambda_0 I$ be diagonal matrices with
elements as the eigenvalues of $K_\rho + \lambda_0 I$ and $K_{\rho_0} + \lambda_0 I$, respectively. Then,

$$
\text{tr} \left\{ (K_\rho + \lambda_0 I)^{-2} (K_{\rho_0} + \lambda_0 I) \right\} - \text{tr} (K_\rho + \lambda_0 I)^{-1}
= \text{tr} \left\{ U \Lambda_\rho^{-2} U' U \Lambda_{\rho_0} U \right\} - \text{tr} \left\{ U \Lambda_\rho^{-1} U \right\}
= \text{tr} \left\{ \Lambda_\rho^{-2} \Lambda_{\rho_0} \right\} - \text{tr} \left\{ \Lambda_\rho^{-1} \right\}
= \frac{1 + (n-1)\rho_0 + \lambda_0}{1 + (n-1)\rho + \lambda_0} + (n-1) \frac{1 - \rho_0 + \lambda_0}{1 - \rho + \lambda_0}
- \frac{1}{1 + (n-1)\rho + \lambda_0} - (n-1) \frac{1}{1 - \rho + \lambda_0}
= (n-1) (\rho - \rho_0) \left( 1 - \frac{1}{(1 - \rho + \lambda_0)^2} - \frac{1}{(1 + (n-1)\rho + \lambda_0)^2} \right)
= (n-1) (\rho - \rho_0) \frac{n \rho (2 + 2\lambda_0 + (n-2)\rho)}{(1 - \rho + \lambda_0)^2 (1 + (n-1)\rho + \lambda_0)^2}.
$$

Thus,

$$
\text{tr} \left\{ (K_\rho + \lambda_0 I)^{-2} (K_{\rho_0} + \lambda_0 I) \right\} - \text{tr} (K_\rho + \lambda_0 I)^{-1} \geq 0 \iff \rho \geq \rho_0,
$$

and hence $E_{\rho_0, \lambda_0} \Gamma(\rho, \lambda_0) \geq 0 \iff \rho \geq \rho_0$. Note that in the above algebra $\lambda_0 > 0$ allows for $\rho_0, \rho \in \{0, 1\}$.

The following result, improves on the previous one by generalizing the type of kernels.

**Result 6.** Suppose, $\kappa_0$ and $\kappa_1$ are kernels such that the eigenvectors of their Gram matrices\(^4\) are same. Then,

$$
E_{\kappa_0, \lambda_0} \Gamma(\kappa_1, \lambda_0) < 0, \text{ for } \lambda(K_1) \prec \lambda(K_0).
$$

**Proof.** Following earlier notations, let $\Lambda_1$ and $\Lambda_0$ be the diagonal matrices of the eigenvalues of $K_1 + \lambda_0 I$ and $K_0 + \lambda_0 I$, respectively; also let $\gamma_{0i} = \lambda_i(K_0)$ and $\gamma_{1i} = \lambda_i(K_1)$.
\[ \lambda_i(K_1). \] Since, by assumption their eigenvectors are same, we have,

\[
\text{tr} \left\{ (K_1 + \lambda_0 I)^{-2} (K_0 + \lambda_0 I) \right\} - \text{tr} (K_1 + \lambda_0 I)^{-1} = \text{tr} \left\{ \Lambda_1^{-2} \Lambda_0 \right\} - \text{tr} \left\{ \Lambda_1^{-1} \right\}
\]

\[ = \sum_{i=1}^{n} \frac{\gamma_{0i} + \lambda_0}{(\gamma_{1i} + \lambda_0)^2} - \sum_{i=1}^{n} \frac{1}{\gamma_{1i} + \lambda_0} \]

For \( \lambda(K_1) \prec \lambda(K_0) \), using (3.2.2) with \( p = 2 \), \( a_i = \gamma_{0i} + \lambda_0 \), and \( b_i = \gamma_{1i} + \lambda_0 \), we get,

\[ \sum_{i=1}^{n} \frac{\gamma_{0i} + \lambda_0}{(\gamma_{1i} + \lambda_0)^2} < \sum_{i=1}^{n} \frac{1}{\gamma_{1i} + \lambda_0}. \]

Hence the proof. \( \Box \)

Of course, the RBF kernel does not have this property (same eigenvector for the two Gram matrices), but the following parametric class of product kernels does. Consider a completely specified kernel \( \kappa_\star \) (normalized, if necessary, so that \( \kappa_\star(x,x) = 1 \) for all \( x \)). Let

\[ K_\rho = \{ \kappa_\star \kappa_\rho : 0 \leq \rho \leq 1 \} \]

where \( \kappa_\rho \) is as defined in Result [5] Let \( K_\star \) and \( K_\rho = (1-\rho)I + \rho J \) be the Gram matrices corresponding to \( \kappa_\star \) and \( \kappa_\rho \) respectively, with \( K_\star = U \Lambda U' \) as the eigendecomposition, then

\[ K_\star \circ K_\rho = U [(1-\rho)I + \rho \Lambda] U'. \]

Hence, the eigenvectors of \( K_\star \circ K_\rho \) and \( K_\star \) are same for all \( \rho \). And thus, the above Result holds for this special class of kernels as well.

Based on the above results for the special cases, the computational evidence from the previous section (see for example, Figures 3.1, 3.3, and 3.5), and the measurement based patterns from the previous chapter (see for example, Figures 2.9, C.19 – C.27 and 2.11), we are confident that the other side of Proposition 1 holds as well.
Conjecture 1. At $\lambda = \lambda_0$, for $\lambda(K_1) \prec \lambda(K_0)$, we have

$$E_{\kappa_0, \lambda_0} \Gamma(\kappa_1, \lambda_0) \leq 0.$$ 

For the RBF kernel, this can be written in the parametric form as:

$$h < h_0 \implies E_{h_0, \lambda_0} \Gamma(h, \lambda_0) \leq 0.$$ 

### 3.4.3 Theoretical Behavior of the Tool Based on Functional Norm

The following result gives a simple threshold for diagnostics based on the expected functional norm defined towards the beginning of Section 3.4.

**Result 7.** For $\kappa_0 = \kappa_1$, $E_{\kappa_0, \lambda_0} \nu(\kappa_0, \lambda_0) < n$.

**Proof.** It follows trivially:

$$E_{\kappa_0, \lambda_0} \nu(\kappa_0, \lambda_0) = \text{tr} \{(K_0 + \lambda_0 I)^{-1} K_0\} = \sum_{i=1}^{n} \frac{\gamma_{0i}}{\gamma_{0i} + \lambda_0} < n,$$

where, $\{\gamma_{0i}\}_{1}^{n}$ are the eigenvalues of $K_0$. \qed

Further, the above bound can be marginally improved as follows;

**Result 8.**

$$\frac{n}{n + \lambda_0} < E_{\kappa, \lambda_0} \nu(\kappa, \lambda_0) < \frac{n}{1 + \lambda_0}.$$ 

**Proof.** Appealing to Lemma 7 and Lemma 5 with $g(x) = \frac{x}{x + \lambda_0}$ as the concave function of $x \in [0, n]$, it is easy to see that for,

$$g^*(K) = E_{\kappa, \lambda_0} \nu(\kappa, \lambda_0) = \sum_{i=1}^{n} \frac{\lambda_i(K)}{\lambda_i(K) + \lambda_0},$$
$g^*(K_1) \geq g^*(K_2)$ whenever $\lambda(K_1) \prec \lambda(K_2)$. Note that, for kernels under consideration, $\lambda(I) \prec \lambda(K) \prec \lambda(J)$. The eigenvalues of $I$ and $J$ are $(1, 1, \ldots, 1)$ and $(n, 0, \ldots, 0)$, respectively; hence,

$$\frac{n}{1 + \lambda_0} = g^*(I) > E_{\kappa, \lambda_0} \nu(\kappa, \lambda_0) > \frac{n}{n + \lambda_0} = g^*(J).$$

The diagnostic heuristics using these bounds is the following: if, $\nu(\kappa_1, \lambda_0)$, is “far away” from these bounds then such a model kernel, $\kappa_1$, can be inferred to be “not near” $\kappa_0$.

Although, as evidenced from Figure 3.6, the sign-change property of $E_{\kappa_0, \lambda_0} \Upsilon(\kappa_1, \lambda_0)$ does not hold for any generic $\lambda_0 > 0$; as a diagnostic tool, $\Upsilon(\kappa_1, \lambda)$, is still pretty useful for small values of $\lambda_0$. Note that, since, $\lambda_0$ is the signal to noise ratio for the model, statistically, values of $\lambda_0 > 1$ are not at all meaningful. Further, the tool based on GCV is only a measure of the prediction performance of the estimated model. Beyond that, if we are interested in the actual function estimation, the the functional norm of the estimated function plays a crucial role. And in situations where the function is “sufficiently distinguishable” from the measurement noise, the computational evidence certainly seems promising enough to use $\Upsilon(\kappa_1, \lambda)$ as a viable diagnostic tool.

### 3.5 Concluding Remarks on the Theoretical Aspects

In this chapter we have explored a few theoretical properties of the diagnostic tool based on GCV and functional norm. We started out with some review of the necessary concepts and results in Section 3.1. The most important tool for our purpose was the result due to Bapat and Sunder (1985) which provided a majorization relationship between the eigenvalue vectors of two Gram matrices. In our context, we were interested in comparing and contrasting the Gram matrix corresponding to
the “true” underlying kernel with the Gram matrix of the model kernel. For this purpose we used the first moment (expectation) of the diagnostic tool which are just the expectations of quadratic forms of the measurements. These only require the existence of the first two moments of the underlying generation process. In this respect, although we restricted ourselves to the framework of Gaussian Process, the results we investigated, are generic to any weekly stationary stochastic process, as long as the prediction was based on linear posterior expectations. The important utility of the Gaussian assumption of the underlying stochastic process, which we hope to investigate in future, is the variability of these diagnostic tools. Similar to the measurement, $Y^{tr}$, based explorations of Chapter 2, we explored their corresponding expectations across $h \times \lambda$ grid and further at $\lambda = \lambda_0$ in Section 3.3. We also proved the monotonicity of the $E_{h,\lambda} \text{GCV}(h, \lambda)$ and $E_{h,\lambda} \nu(h, \lambda)$ which were observed during the initial explorations in Figures 2.5, 2.6, and B.1 – B.18.

Then, equipped with the majorization tool, in Section 3.4, we moved away from the parametric representation of the kernel functions and we could make very generic statements using a non-parametric representation of the kernel functions. In Section 3.4.1 we showed, in a classical regression framework, how to reduce the search space for the model regularization coefficient using the magnitude and sign of a term which involves the “discrepancy-decorrelation” matrix that was introduced in Section 2.4.1. For the diagnostic tool based on GCV, we proved that under the true distribution, its expectation is bounded below by 0 when the eigenvalue vector of the true Gram matrix is majorized by that of the model Gram matrix. Under special cases of simple family of kernels we proved that it is also bounded above by 0 when the other direction of the majorization holds. This validates the patterns observed in Chapter 2 for the
tool based on GCV, at least for simple kernels. Based on these, we conjectured that for general kernels, the other direction should hold as well. The tool based on the functional norm is a useful diagnostic tool in most statistically meaningful situations with low signal to noise ratio, but as an algebraic concept it does not enjoy the similar bounds as that of the tool based on GCV.
Chapter 4: Conclusion and Future Work

Finally, in this brief chapter we review the various aspects of our research and provide some avenues for future work. To begin with let us consider a result which we think should help in proving Conjecture 1 and also be utilized to develop a more generic framework to study higher moments of the diagnostic tools. Additionally, using this result we also show that the “decorrelation-discrepancy” matrix, defined in (3.4.1), can itself be used as a diagnostic tool.

4.1 Comments on Conjecture 1

Recall that Proposition 1 in Section 3.4.2 states that the expected value of the diagnostic tool based on GCV, $E_{\nu_0, \lambda_0} \Gamma(h, \lambda_0)$, is bounded below by 0 when the eigenvalue vector of the Gram matrix corresponding to the true kernel is majorized by that of the model kernel, that is, $\lambda(K_0) \prec \lambda(K_1)$. And based on the various computer simulation experiments that we have explored, for the tools using the measurements, $Y^{tr}$, (see Chapter 2) as well as for their expected values (see Section 3.3), we expect that Conjecture 1 should hold as well.

Note that Proposition 1, Lemma 6 and Lemma 7 are valid for all positive-definite matrices. However, all our computer simulation experiments were restricted to Gaussian kernels, where the kernel function is non-negative. So, in our first approach to
prove Conjecture \[1\] we intend to restrict to only non-negative kernels. In this respect we consider the following result.

Let \( A = [a_{ij}] \) and \( B = [b_{ij}] \) be two \( n \times n \) real symmetric positive definite matrices such that, \( a_{ii} = b_{ii} = 1 \) for all \( i = 1, \ldots, n \). Further assume that \( a_{ij} > 0, b_{ij} > 0 \) for all \( i \neq j \in \{1, \ldots, n\} \), that is, both the matrices have non-negative elements. Note that, in this case, \( A - A \circ B \), has all positive off-diagonal entries as well, that is, \( a_{ij} > a_{ij} b_{ij} \) for all \( i \neq j \).

**Lemma 8.** Let \( H = (A \circ B)^{-1}A \) where \( A \) and \( B \) are as described above, then \( 0 \leq h_{ij} < 1 \) with \( h_{ii} > 0 \) for all \( i, j \in \{1, \ldots, n\} \). Thus, the matrix \( n^{-1}H \) is a sub-stochastic matrix and for a suitable \( s > 0 \), the matrix \( s I - H \) is a \( \mathbf{M} \)-matrix.

**Proof.** Since \( a_{ij} b_{ij} \leq a_{ij} \), the cone generated by columns of \( A \circ B \) contains the cone generated by columns of \( A \). So, there exists a \( H \) such that \((A \circ B)H = A\) where \( h_{ij} \geq 0 \) for all \( i, j \). In fact, since \( a_{ii} = b_{ii} = 1 \) for all \( i \), necessarily, we have \( 0 < h_{ii} < 1 \). Therefore, \( \text{tr}(H) < n \). Further, since both \( A \) and \( A \circ B \) are p.d. matrices, \( H = (A \circ B)^{-1}A \) is p.d. as well. Since for a p.d. matrix all its principle minors are p.d, it follows that \( h_{ij} < 1 \) for all \( i \neq j \). This implies that all the row sums and column sums of \( H \) are less than \( n \); hence, \( n^{-1}H \) is a sub-stochastic matrix. Also, trivially, we can choose \( s > n^{-1} \max\{h_{ii}, i = 1, \ldots, n\} \) so that \( s I - n^{-1}H \) is a \( \mathbf{M} \)-matrix. \( \square \)

As discussed in Section \[3.4.1\] we can use the expression defined in \((3.4.1)\) itself as a diagnostic tool, utilizing its sign-change properties. The proposition below establishes this property for its expected value.

\[5\]a \( \mathbf{M} \)-matrix is of the form \( sI - R \) where \( r_{ij} \geq 0 \) and \( s > \rho(R) \), see, e.g., Berman and Plemons \[1979\] p. 133

\[6\]\{\( A\alpha : \alpha \geq 0 \}\}, the set of all non-negative linear combinations of the columns of \( A \) is the cone generated by the columns of \( A \), see Berman and Plemons \[1979\] Sec. 1.2

95
Proposition 2. For $\lambda = \lambda_0$, consider $\Xi(\kappa_1, \lambda_0)$ as defined in (3.4.1). If $\kappa_1 = \kappa_0 \cdot \kappa_*$, for some unknown kernel $\kappa_*$, then, $E_{\kappa_0, \lambda_0} \Xi(\kappa_1, \lambda_0) < 0$. Otherwise, if, $\lambda(\mathbf{K}_0) \prec \lambda(\mathbf{K}_1)$, then $E_{\kappa_0, \lambda_0} \Xi(\kappa_1, \lambda_0) \geq 0$.

Proof. When $\kappa_1 = \kappa_0 \cdot \kappa_*$, for the corresponding Gram matrices, we can write $\mathbf{K}_1 = \mathbf{K}_0 \circ \mathbf{K}_*$. Now, let $A = (1 + \lambda_0)^{-1}(\mathbf{K}_0 + \lambda_0 \mathbf{I})$, $B = \mathbf{K}_*$ so that

$$A \circ B = \frac{1}{1 + \lambda_0} (\mathbf{K}_0 + \lambda_0 \mathbf{I}) \circ \mathbf{K}_* = \frac{1}{1 + \lambda_0} (\mathbf{K}_0 \circ \mathbf{K}_* + \lambda_0 \mathbf{I}) = \frac{1}{1 + \lambda_0} (\mathbf{K}_1 + \lambda_0 \mathbf{I}).$$

Then, $(A \circ B)^{-1}A = (\mathbf{K}_1 + \lambda_0 \mathbf{I})^{-1}(\mathbf{K}_0 + \lambda_0 \mathbf{I})$, and using Lemma 8 we get,

$$E_{\kappa_0, \lambda_0} \Xi(\kappa_1, \lambda_0) = \text{tr} \left\{ (\mathbf{K}_1 + \lambda_0 \mathbf{I})^{-1}(\mathbf{K}_0 + \lambda_0 \mathbf{I}) \right\} < n.$$

In the other case, when $\lambda(\mathbf{K}_0) \prec \lambda(\mathbf{K}_1)$, the proof is very similar to that of Proposition 1. \qedsymbol

Now, going back to the tool based on GCV, using Lemma 8, the properties of $M$-matrices, and the fact that $\text{tr} \{ AB \}$ can be considered as an inner product in $\mathbb{R}^{n^2}$ dimensional space between $\text{vec}(A \circ B)$ and $\mathbf{1}$, Conjecture 1 can be stated in the following way:

For a convex decreasing function, $g$,

$$\text{tr} \left\{ g(A \circ B) \left( (A \circ B)^{-1}A - I \right) \right\} \leq 0.$$

For the full generality of all legitimate kernel functions, which are not necessarily non-negative, the cone generated by columns of $A$ may not be contained in the cone generated by columns of $A \circ B$. Therefore, this approach will not work.

$^7$vec($M$) represents the matrix $M$ as a $n^2$ dimensional vector.
4.2 Concluding Remarks and Future work

The issue of “learning the kernel” is an interesting problem from the point of view of understanding the data generation process. The essential contribution of this work was to show that in the presence of an underlying data generation process, through appropriate diagnostic tools, we can reduce the computational effort and search space for learning the best collection of kernels.

The main diagnostic tool developed for this purpose was the difference of the generalized cross-validation with its expected value under the modeling distribution (see Section 2.4). This difference is computable from the data. Through various computer simulation experiments (see Sections 2.4.3 - 2.4.5) we observed that this difference changes sign “near” the true kernel bandwidth. This observation was later, partially validated, for one-side, in Proposition 1 where we showed that the expected value of this difference, under the true underlying distribution, is in fact bounded below by 0. Under some special cases of kernels we showed that, for the other side, it is bounded above by 0 (see Results 5 and 6). Based on these results and earlier simulations in Section 3.3 we conjectured for the other side that, in the general case of positive definite kernels, the expected difference is bounded above by 0 (see Conjecture 1). In the previous section, through Lemma 8 we provided a possible means to prove this conjecture for the special case of non-negative kernels. An interesting aspect to note here is that the statements in both Proposition 1 and Conjecture 1 are not asymptotic, that is, they are based on finite number of training samples, which we hope is a useful contribution to the literature.

The other diagnostic tool considered was the difference of the functional norm of the estimated function with its expected value under the modeling distribution. The
experiments done in Sections 2.4 and 3.3 suggest that, under meaningful values of the signal to noise ratio, this is viable tool as well. We expect that it is possible to have analogous results for this tool by restricting to meaningful signal to noise ratios. It is certainly an avenue we intend to pursue ahead.

Further, through Proposition 2 we showed decorrelated measurements also provide useful diagnostic information regarding the underlying distribution.

Although, in analysis of most real data, the assumption of the existence of an underlying data generation process, may be not always hold, we believe this work provides quantitative justification towards the choice of a better kernel.

There are quite a few things that we intend to address in future works. Among them the most important one is to complete the proof for Conjecture 1 along the line of argument discussed in the previous section first. Along the way we hope to employ a more generic technique which can be utilized to prove the similar results, motivated by the observations made in Section 2.4.5 for the variability of the measured tools. To understand the variability from this perspective, we need to carefully utilize the distributional assumptions. While studying the expectations required only the existence of the first two moments, the variance of these tools requires fourth moments. It is here that the Gaussian assumption of the underlying process, we expect, will be most useful.

Another aspect of the current work that we intend to pursue is the marginal asymmetry of the two diagnostic tools with respect to $\lambda$. Of interest also is to investigate other possible diagnostics, such as, validation set MSE, regularization cost, possibly a combination of the GCV and the functional norm, etc.
Further, we plan to perform computer experimentation with kernels, other than RBF, initially being closed under the Hadamard product operation. Then, subsequently, to investigate more general families of kernels outside the Hadamard product, but within the framework of convexity, possibly leveraging the work in semidefinite programming.

Using these diagnostics, we also intend to apply the best fitting kernel in the RVM framework to investigate the sparseness of the selected kernel bases.
A.1 GCV and Functional norm on $h \times \lambda$ grid using multiple replications of the measurements

This section provides the supporting Figures referred to in Section 2.2.3

Figure A.1: GCV (2.2.1): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y^{tr}_2$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal).
Figure A.2: GCV (2.2.1): GCV$(h, \lambda)$ on a $h \times \lambda$ grid using $Y_3^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log$(h_0)$ (vertical) and log$(\lambda_0)$ (horizontal).

Figure A.3: GCV (2.2.1): GCV$(h, \lambda)$ on a $h \times \lambda$ grid using $Y_3^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log$(h_0)$ (vertical) and log$(\lambda_0)$ (horizontal).
Figure A.4: GCV \([2.2.1]\): \(\text{GCV}(h, \lambda)\) on a \(h \times \lambda\) grid using \(Y_5^{fr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines are at \(\log(h_0)\) (vertical) and \(\log(\lambda_0)\) (horizontal).

Figure A.5: GCV \([2.2.1]\): \(\text{GCV}(h, \lambda)\) on a \(h \times \lambda\) grid using \(Y_6^{fr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines are at \(\log(h_0)\) (vertical) and \(\log(\lambda_0)\) (horizontal).
$h_0 = 3 \quad h_0 = 14.6 \quad h_0 = 26.1$

Figure A.6: GCV \([2.2.1]\): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y^fr_7$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).

$\log(\text{mean.rkhs.norm})$

$h_0 = 14.0 \quad h_0 = 26.1$

Figure A.7: GCV \([2.2.1]\): GCV($h, \lambda$) on a $h \times \lambda$ grid using $Y^fr_8$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).
Figure A.8: GCV [2.2.1]: GCV(\(h, \lambda\)) on a \(h \times \lambda\) grid using \(Y^\text{tr}_0\). Top row: 1.5 \(\leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines are at \(\log(h_0)\) (vertical) and \(\log(\lambda_0)\) (horizontal).

Figure A.9: GCV [2.2.1]: GCV(\(h, \lambda\)) on a \(h \times \lambda\) grid using \(Y^\text{tr}_{10}\). Top row: 1.5 \(\leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines are at \(\log(h_0)\) (vertical) and \(\log(\lambda_0)\) (horizontal).
Figure A.10: Functional norm $\log \nu(h, \lambda)$ on a $h \times \lambda$ grid using $Y_2^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).
Figure A.11: Functional norm $\| \nabla (h, \lambda) \|$ on a $h \times \lambda$ grid using $Y^{tr}_3$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal).

Figure A.12: Functional norm $\| \nabla (h, \lambda) \|$ on a $h \times \lambda$ grid using $Y^{tr}_4$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at log($h_0$) (vertical) and log($\lambda_0$) (horizontal).
Figure A.13: Functional norm of : \( \log \nu(h, \lambda) \) on a \( h \times \lambda \) grid using \( Y_6^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines are at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal).

Figure A.14: Functional norm of : \( \log \nu(h, \lambda) \) on a \( h \times \lambda \) grid using \( Y_6^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines are at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal).
Figure A.15: Functional norm $\log \nu (h, \lambda)$ on a $h \times \lambda$ grid using $Y^TR$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).

Figure A.16: Functional norm $\log \nu (h, \lambda)$ on a $h \times \lambda$ grid using $Y^TR$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines are at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal).
Figure A.17: Functional norm $\log(\text{rkhs.norm})$: $\log \nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{0}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_{0}/2 \leq h \leq 2h_{0}$. Dotted lines are at $\log(h_{0})$ (vertical) and $\log(\lambda_{0})$ (horizontal).

Figure A.18: Functional norm $\log(\text{rkhs.norm})$: $\log \nu(h,\lambda)$ on a $h \times \lambda$ grid using $Y_{0}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_{0}/2 \leq h \leq 2h_{0}$. Dotted lines are at $\log(h_{0})$ (vertical) and $\log(\lambda_{0})$ (horizontal).
\[ h_0 = 3 \quad \text{\textit{h0 = 3}} \]
\[ h_0 = 14.6 \quad \text{\textit{h0 = 14.6}} \]
\[ h_0 = 26.1 \quad \text{\textit{h0 = 26.1}} \]

Figure A.19: Optimal cost \( C(h, \lambda) \) on a \( h \times \lambda \) grid using \( Y_1^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines are at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal).

### A.2 Optimal cost and Validation set prediction on \( h \times \lambda \) grid using \( Y_1^{tr} \)

This section provides the supporting Figures referred to in Section \( \text{\textit{2.2.3}} \) related to the optimal cost function \( C(h, \lambda) \) and the out-of-sample predictions, \( \text{MSE}_v(h, \lambda) \). Both these plots are based the measurements \( Y_1^{tr} \). Similar patterns were observed for other replicated measurements. Since, we do not investigate these two measures in much detail in this thesis, figures corresponding to other measurements are not shown.

110
Figure A.20: Validation set predictions \[(2.2.5)\]: MSE \(h, \lambda \) on a \(h \times \lambda \) grid using \(Y_1^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines are at \(\log(h_0)\) (vertical) and \(\log(\lambda_0)\) (horizontal).

### A.3 GCV and Functional norm at \(\lambda = \lambda_0\) using multiple replications of the measurements

This section provides the supporting Figures referred to in Section \[2.2.4\].
Figure A.21: GCV $\{2.2.1\}$ at $\lambda = \lambda_0$: GCV$(h, \lambda_0)$ as a function of $\log(h)$ using $Y_2^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.

Figure A.22: GCV $\{2.2.1\}$ at $\lambda = \lambda_0$: GCV$(h, \lambda_0)$ as a function of $\log(h)$ using $Y_3^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.  

112
Figure A.23: GCV \([2.2.1]\) at \(\lambda = \lambda_0\): GCV\((h, \lambda_0)\) as a function of \(\log(h)\) using \(Y_4^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at \(\log(h_0)\).

Figure A.24: GCV \([2.2.1]\) at \(\lambda = \lambda_0\): GCV\((h, \lambda_0)\) as a function of \(\log(h)\) using \(Y_5^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at \(\log(h_0)\).
Figure A.25: GCV [2.2.1] at $\lambda = \lambda_0$: GCV($h, \lambda_0$) as a function of log($h$) using $Y_{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$).

Figure A.26: GCV [2.2.1] at $\lambda = \lambda_0$: GCV($h, \lambda_0$) as a function of log($h$) using $Y_{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$).
Figure A.27: GCV \([2.2.1]\) at \(\lambda = \lambda_0\): GCV\((h, \lambda_0)\) as a function of log\((h)\) using \(Y_{9tr}^{\text{tr}}\).
Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at log\((h_0)\).

Figure A.28: GCV \([2.2.1]\) at \(\lambda = \lambda_0\): GCV\((h, \lambda_0)\) as a function of log\((h)\) using \(Y_{9tr}^{\text{tr}}\).
Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at log\((h_0)\).
Figure A.29: GCV \( \{2.2.1\} \) at \( \lambda = \lambda_0 \): GCV\((h, \lambda_0)\) as a function of \(\log(h)\) using \(Y_{10}^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at \(\log(h_0)\).

Figure A.30: Validation set predictions \( \{2.2.5\} \) at \( \lambda = \lambda_0 \): MSE\(_v\)(\(h, \lambda_0\)) as a function of \(\log(h)\) using \(Y_{10}^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines are at \(\log(h_0)\).
Figure A.31: Functional norm (2.2.2) at $\lambda = \lambda_0$: log $\nu(h, \lambda_0)$ as a function of log($h$) using $Y_2^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$) (vertical) and log($n$) \approx 5.3 (horizontal).

Figure A.32: Functional norm (2.2.2) at $\lambda = \lambda_0$: log $\nu(h, \lambda_0)$ as a function of log($h$) using $Y_3^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$) (vertical) and log($n$) \approx 5.3 (horizontal).
Figure A.33: Functional norm $\log \nu(h, \lambda_0)$ at $\lambda = \lambda_0$: as a function of $\log(h)$ using $Y_{4fr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$ (vertical) and $\log(n) \approx 5.3$ (horizontal).

Figure A.34: Functional norm $\log \nu(h, \lambda_0)$ at $\lambda = \lambda_0$: as a function of $\log(h)$ using $Y_{5fr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$ (vertical) and $\log(n) \approx 5.3$ (horizontal).
Figure A.35: Functional norm \([2.2.2]\) at \(\lambda = \lambda_0\): \(\log \nu(h, \lambda_0)\) as a function of \(\log(h)\) using \(\mathcal{Y}_6^tr\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at \(\log(h_0)\) (vertical) and \(\log(n) \approx 5.3\) (horizontal).

Figure A.36: Functional norm \([2.2.2]\) at \(\lambda = \lambda_0\): \(\log \nu(h, \lambda_0)\) as a function of \(\log(h)\) using \(\mathcal{Y}_7^tr\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at \(\log(h_0)\) (vertical) and \(\log(n) \approx 5.3\) (horizontal).
Figure A.37: Functional norm \([2.2.2]\) at \(\lambda = \lambda_0\): \(\log \nu(h, \lambda_0)\) as a function of \(\log(h)\) using \(Y_{8tr}^r\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at \(\log(h_0)\) (vertical) and \(\log(n) \approx 5.3\) (horizontal).

Figure A.38: Functional norm \([2.2.2]\) at \(\lambda = \lambda_0\): \(\log \nu(h, \lambda_0)\) as a function of \(\log(h)\) using \(Y_{9tr}^r\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted line is at \(\log(h_0)\) (vertical) and \(\log(n) \approx 5.3\) (horizontal).
Figure A.39: Functional norm \( \log (\nu(h, \lambda)) \) at \( \lambda = \lambda_0 \) as a function of \( \log(h) \) using \( Y_{10}^{\text{f}} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted line is at \( \log(h_0) \) (vertical) and \( \log(n) \approx 5.3 \) (horizontal).
Appendix B: Supporting Figures for Chapter 2 (Part II)

B.1 Comparing GCV and Functional norm with their respective expectations under the model space at $\lambda = \lambda_0$ using multiple replications of the measurements

This section provides the supporting Figures referred to in Section 2.3.

Figure B.1: Comparing $GCV(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{2tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.  

122
Figure B.2: Comparing $\text{GCV}(h, \lambda_0)$ (solid blue curve) and $\text{E}_{h,\lambda_0}\text{GCV}(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_3^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.

Figure B.3: Comparing $\text{GCV}(h, \lambda_0)$ (solid blue curve) and $\text{E}_{h,\lambda_0}\text{GCV}(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_4^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.
Figure B.4: Comparing GCV($h, \lambda_0$) (solid blue curve) and $E_{h,\lambda_0}$GCV($h, \lambda_0$) (dashed red curve) as a function of log($h$) using $Y_{6tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$).

Figure B.5: Comparing GCV($h, \lambda_0$) (solid blue curve) and $E_{h,\lambda_0}$GCV($h, \lambda_0$) (dashed red curve) as a function of log($h$) using $Y_{6tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at log($h_0$).
Figure B.6: Comparing $\text{GCV}(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\text{GCV}(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.

Figure B.7: Comparing $\text{GCV}(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\text{GCV}(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{tr}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.
Figure B.8: Comparing GCV($h, \lambda_0$) (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{10}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.

Figure B.9: Comparing GCV($h, \lambda_0$) (solid blue curve) and $E_{h,\lambda_0}GCV(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{10}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. 

126
Figure B.10: Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h, \lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_2^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.

Figure B.11: Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h, \lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_3^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. 

127
Figure B.12: Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_4^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.

Figure B.13: Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_5^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. 

128
Figure B.14: Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y^t_t$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.

Figure B.15: Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y^t_t$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. 

129
Figure B.16: Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_9^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$.

Figure B.17: Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h,\lambda_0}\nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_9^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. 

130
Figure B.18: Comparing $\nu(h, \lambda_0)$ (solid blue curve) and $E_{h, \lambda_0} \nu(h, \lambda_0)$ (dashed red curve) as a function of $\log(h)$ using $Y_{10}^g$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted line is at $\log(h_0)$. 
Appendix C: Supporting Figures for Chapter 2 (Part III)

C.1 Visualizing Diagnostic Tools based on GCV and Functional norm on $h \times \lambda$ grid using replicated measurements

This section provides the supporting Figures referred to in Section 2.4.3.

Figure C.1: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_2^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.
Figure C.2: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{3tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). Solid curve (approximately) denotes the sign change boundary.

Figure C.3: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{4tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). Solid curve (approximately) denotes the sign change boundary.
Figure C.4: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{6}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_{0}/2 \leq h \leq 2h_{0}$. Dotted lines at $\log(h_{0})$ (vertical) and $\log(\lambda_{0})$ (horizontal). Solid curve (approximately) denotes the sign change boundary.

Figure C.5: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{6}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_{0}/2 \leq h \leq 2h_{0}$. Dotted lines at $\log(h_{0})$ (vertical) and $\log(\lambda_{0})$ (horizontal). Solid curve (approximately) denotes the sign change boundary.
Figure C.6: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.

Figure C.7: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{str}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.
Figure C.8: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{\theta}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). Solid curve (approximately) denotes the sign change boundary.

Figure C.9: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda)$ on $h \times \lambda$ grid using $Y_{\theta}^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at log($h_0$) (vertical) and log($\lambda_0$) (horizontal). Solid curve (approximately) denotes the sign change boundary.
Figure C.10: Diagnostic tool based on functional norm (2.4.2); $\Upsilon(h, \lambda)$ on $h \times \lambda$ grid using $Y_2^{\mu r}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.
Figure C.11: Diagnostic tool based on functional norm (2.4.2): \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_3^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2 h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary.

Figure C.12: Diagnostic tool based on functional norm (2.4.2): \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y_4^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2 h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary.
Figure C.13: Diagnostic tool based on functional norm (2.4.2): $\mathcal{Y}(h, \lambda)$ on $h \times \lambda$ grid using $Y_{5tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.

Figure C.14: Diagnostic tool based on functional norm (2.4.2): $\mathcal{Y}(h, \lambda)$ on $h \times \lambda$ grid using $Y_{6tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.
Figure C.15: Diagnostic tool based on functional norm (2.4.2): \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y\text{ }^\text{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary.

Figure C.16: Diagnostic tool based on functional norm (2.4.2): \( \Upsilon(h, \lambda) \) on \( h \times \lambda \) grid using \( Y\text{ }^\text{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and \( \log(\lambda_0) \) (horizontal). Solid curve (approximately) denotes the sign change boundary.
Figure C.17: Diagnostic tool based on functional norm (2.4.2): $\Upsilon(h, \lambda)$ on $h \times \lambda$ grid using $Y_{00}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.

Figure C.18: Diagnostic tool based on functional norm (2.4.2): $\Upsilon(h, \lambda)$ on $h \times \lambda$ grid using $Y_{10}^{fr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and $\log(\lambda_0)$ (horizontal). Solid curve (approximately) denotes the sign change boundary.
C.2 Visualizing Diagnostic Tools based on GCV and Functional norm at $\lambda = \lambda_0$ using replicated measurements

This section provides the supporting Figures referred to in Section 2.4.4.

$h_0 = 3$  $h_0 = 14.6$  $h_0 = 26.1$

![Graphs](image)

Figure C.19: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y^fr_2$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
Figure C.20: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y^3_{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

Figure C.21: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y^r_{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
Figure C.22: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of log($h$) using $Y_5^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at log($h_0$) (vertical) and 0 (horizontal).

Figure C.23: Diagnostic tool based on GCV (2.4.1): $\Gamma(h, \lambda_0)$ as a function of log($h$) using $Y_6^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at log($h_0$) (vertical) and 0 (horizontal).
Figure C.24: Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_i^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

Figure C.25: Diagnostic tool based on GCV [2.4.1]: $\Gamma(h, \lambda_0)$ as a function of $\log(h)$ using $Y_i^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
Figure C.26: Diagnostic tool based on GCV \([2.4.1]\): \(\Gamma(h, \lambda_0)\) as a function of \(\log(h)\) using \(Y_9^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines at \(\log(h_0)\) (vertical) and 0 (horizontal).

Figure C.27: Diagnostic tool based on GCV \([2.4.1]\): \(\Gamma(h, \lambda_0)\) as a function of \(\log(h)\) using \(Y_{16}^{tr}\). Top row: \(1.5 \leq h \leq 60\); bottom row: \(h_0/2 \leq h \leq 2h_0\). Dotted lines at \(\log(h_0)\) (vertical) and 0 (horizontal).
Figure C.28: Diagnostic tool based on functional norm (2.4.2): $\mathcal{Y}(h, \lambda_0)$ as a function of log$(h)$ using $Y_2^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at log$(h_0)$ (vertical) and 0 (horizontal).
Figure C.29: Diagnostic tool based on functional norm (2.4.2): \( \mathcal{Y}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_3^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal).

Figure C.30: Diagnostic tool based on functional norm (2.4.2): \( \mathcal{Y}(h, \lambda_0) \) as a function of \( \log(h) \) using \( Y_4^{tr} \). Top row: \( 1.5 \leq h \leq 60 \); bottom row: \( h_0/2 \leq h \leq 2h_0 \). Dotted lines at \( \log(h_0) \) (vertical) and 0 (horizontal).
Figure C.31: Diagnostic tool based on functional norm (2.4.2): $Y(h, \lambda_0)$ as a function of $\log(h)$ using $Y_5^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

Figure C.32: Diagnostic tool based on functional norm (2.4.2): $Y(h, \lambda_0)$ as a function of $\log(h)$ using $Y_6^{tr}$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
Figure C.33: Diagnostic tool based on functional norm (2.4.2): $\mathcal{Y}(h, \lambda_0)$ as a function of $\log(h)$ using $\mathbf{Y}_{tr}^r$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

Figure C.34: Diagnostic tool based on functional norm (2.4.2): $\mathcal{Y}(h, \lambda_0)$ as a function of $\log(h)$ using $\mathbf{Y}_{tr}^s$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
Figure C.35: Diagnostic tool based on functional norm (2.4.2): $Y(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^0$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).

Figure C.36: Diagnostic tool based on functional norm (2.4.2): $Y(h, \lambda_0)$ as a function of $\log(h)$ using $Y_{tr}^0$. Top row: $1.5 \leq h \leq 60$; bottom row: $h_0/2 \leq h \leq 2h_0$. Dotted lines at $\log(h_0)$ (vertical) and 0 (horizontal).
Bibliography


157


fitting (in Japanese).” 152, 12–18.

2323.

*Journal of Machine Learning Research*, 1, 211–244.


*Linear Algebra and its Applications*, 223, 731–744.


ematics.

— (1999). *Support vector machines, reproducing kernel Hilbert spaces, and randomized
GACV*, *Advances in kernel methods: support vector learning*, 69–88. MIT Press,
Cambridge, MA.


