INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700  800/521-0600
RICE UNIVERSITY

Non-Parametric Density Contour Estimation

by

Mark A. Gebert

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

APPROVED. THESIS COMMITTEE:

Dennis D. Cox, Chair
Professor of Statistics

David W. Scott
Professor of Statistics

Benham Aazhang
Professor of Electrical Engineering

Houston, Texas

April, 1998
Abstract

Non-Parametric Density Contour Estimation

by

Mark A. Gebert

Estimation of the level sets for an unknown probability density is done with no specific assumed form for that density, that is, non-parametrically. Methods for tackling this problem are presented. Earlier research showed existence and properties of an estimate based on a kernel density estimate in one dimension. Monte Carlo methods further demonstrated the reasonability of extending this approach to two dimensions. An alternative procedure is now considered that focuses on properties of the contour itself; procedures wherein we define and make use of an objective function based on the characterization of contours as enclosing regions of minimum area given a constraint on probability. Restricting our attention to (possibly non-convex) polygons as candidate contours, numeric optimization of this difficult non-smooth objective function is accomplished using pdsopt, for Parallel Direct Search OPTimization, a set of routines developed for minimization of a scalar-valued function over a high-dimensional domain. Motivation for this
method is given, as well as results of simulations done to test it; these include exploration of a Lagrange-multiplier penalty on area and the need which arises for addition of a penalty on the "roughness" of a polygonal contour.
Acknowledgments

I would like to thank my wonderful wife, Frieda, without whose urging I would have never again taken on the mantle of student.

I also want to acknowledge a great debt to my advisor, Dennis Cox, who guided me ably toward my goal.

The other members of my committee, David Scott and Benham Aazhang, and the other graduate students in Statistics at Rice University deserve a “Thank You” for their reviews and inputs on my work. The faculty in the Division of Mathematical and Information Sciences at Sam Houston State University played an important role in this as well: they allowed me to work a schedule that gave me time to complete this, and they paid me so that I and my children could continue to eat.
## Contents

Abstract .............................................................. ii
Acknowledgments ...................................................... iv
List of Illustrations .................................................. viii
List of Tables ......................................................... xix

1 Introduction ......................................................... 1

2 Basic Notions & Literature Review ............................. 4

3 Kernel-Based Density Contour Estimates ....................... 8
   3.1 Kernel-Based Contour Estimation in One Dimension ........ 8
       3.1.1 Theory .................................................. 8
       3.1.2 Discussion ............................................. 14
       3.1.3 Simulation ............................................. 16
   3.2 Kernel-Based Contour Estimation in Higher Dimensions .... 23
       3.2.1 Theory .................................................. 23
       3.2.2 Simulation ............................................. 23

4 Polygon Optimization-Based Contour Estimation ............ 34
   4.1 Description of Optimization ................................. 34
4.2 Algorithm ......................................................... 35
  4.2.1 Computation of Probability Content and Area ............... 37
  4.2.2 Non-Smoothness of the Objective Function .................. 39
4.3 Simulation Results .................................................. 44
  4.3.1 Effect of Penalties, \(\lambda_i\), and Number of Vertices ...... 45
  4.3.2 Finding a Contour of Desired Probability Content .......... 52
  4.3.3 Effect of Different Starting Guesses ....................... 63
  4.3.4 Comparison to Kernel-Based Density Contour Estimates ... 78
  4.3.5 Behavior with Asymmetric/Bimodal Data .................... 90

5 Concluding Remarks ............................................... 108

6 Proofs of Theorems .................................................. 110
  6.1 Properties of \(\hat{f}_n\) ........................................... 110
  6.2 Royden exercise .................................................. 111
  6.3 One-Dimensional Density Contour Estimation ................. 112
  6.4 Asymptotic Normality of \(\hat{f}_n(x_\alpha)\) ....................... 114
  6.5 Basis Functions for a Rotationally Invariant Process ....... 114

A Symbols Used ..................................................... 116

B Karhunen-Loève Expansion ......................................... 117
B.1 Karhunen-Loève Theorem .......................... 117
B.2 Karhunen-Loève Expansion of a Gaussian Process ............ 118

C Figures Illustrating Optimization-Based Contours 119

C.1 Contours for Range of Penalties on Area and Perimeter ....... 119
C.2 Contours for Various Numbers of Vertices, Penalties on Perimeter . 131
  C.2.1 Different Probability Contents ............................. 132
  C.2.2 Different Penalties on Perimeter ......................... 144
  C.2.3 Different Numbers of Vertices ............................. 156
C.3 Penalties on Area for Desired Probabilities ..................... 165
C.4 Contours for Different Random Data Sets ......................... 173

Bibliography 199
Illustrations

3.1 QQ-Plot of $\hat{x}_{na}$’s from 100 Trials of 100 I.I.D. Replications Each
and Fixed, Optimal Bandwidth ........................................ 18

3.2 QQ-Plot of $\hat{x}_{na}$’s from 100 Trials of 100 I.I.D. Replications Each
and BCV-Determined Bandwidth ........................................ 19

3.3 QQ-Plot of $\hat{x}_{na}$’s from 100 Trials of 1000 I.I.D. Replications Each
and Fixed, Optimal Bandwidth .......................................... 20

3.4 QQ-Plot of $\hat{x}_{na}$’s from 100 Trials of 1000 I.I.D. Replications Each
and BCV-Determined Bandwidth .......................................... 21

3.5 Mean of DFT Coefficients 2-40 for Circular Contour ................. 26

3.6 Variance of DFT Coefficients 1-40 for Circular Contour ............ 27

3.7 QQ-Plots for Standardized DFT Coefficients 1-9 for Circular
Contour ................................................................. 29

3.8 True ($\alpha = .03$) Diamond Contour ................................ 30

3.9 Mean of DFT Coefficients 2-40 for Diamond Contour ............... 31

3.10 Variance of DFT Coefficients 1-40 Diamond Contour ............. 32

3.11 QQ-Plots for Standardized DFT Coefficients 1-9 for Diamond
Contour ................................................................. 33
4.1 Proportion of Point Deemed Interior to Polygon .......................... 43

4.2 Probability versus Number of Vertices ................................. 50

4.3 Objective Function versus Number of Vertices ........................ 51

4.4 Penalty on Perimeter= 0.0, Number of Vertices= 16—Solid Line,
25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
Contour ......................................................... 53

4.5 Penalty on Perimeter= 0.0, Number of Vertices= 7—Solid Line,
25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
Contour ......................................................... 55

4.6 Penalty on Perimeter= 0.04, Number of Vertices= 7—Solid Line,
25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
Contour ......................................................... 56

4.7 Penalty on Perimeter= 0.04, Number of Vertices= 16—Solid Line,
25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
Contour ......................................................... 57

4.8 Number of Vertices= 13—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02;
Dashed Line, 0.04 ........................................ 59

4.9 Penalty on Perimeter= 0.04—Solid Line, $\nu = 7$; Dotted Line, 10;
Short-Dashed Line, 13; and Long-Dashed Line, 16 ..................... 61

4.10 End Contours with Different Starting Radii, n=100—Solid Line,
r = 3.0; Dotted, r = 1.3; Dashed, r = 0.6 .......................... 67
4.11 End Contours with Different Starting Radii, \( n = 200 \)—Solid Line,
\[ r = 3.0; \text{ Dotted, } r = 1.3; \text{ Dashed, } r = 0.6 \] 68

4.12 End Contours with Different Starting Radii, \( n = 500 \)—Solid Line,
\[ r = 3.0; \text{ Dotted, } r = 1.3; \text{ Dashed, } r = 0.6 \] 69

4.13 End Contours with Different Starting Radii, \( n = 1000 \)—Solid Line,
\[ r = 3.0; \text{ Dotted, } r = 1.3; \text{ Dashed, } r = 0.6 \] 71

4.14 End Contours with Different Starting Centers, \( n = 100 \)—Solid Line, \((1.0, 0); \text{ Dotted Line, } (0.5, 0); \text{ Dashed Line, } (0, 0)\) 73

4.15 End Contours with Different Starting Centers, \( n = 200 \)—Solid Line, \((1.0, 0); \text{ Dotted Line, } (0.5, 0); \text{ Dashed Line, } (0, 0)\) 75

4.16 End Contours with Different Starting Centers, \( n = 500 \)—Solid Line, \((1.0, 0); \text{ Dotted Line, } (0.5, 0); \text{ Dashed Line, } (0, 0)\) 76

4.17 End Contours with Different Starting Centers, \( n = 1000 \)—Solid Line, \((1.0, 0); \text{ Dotted Line, } (0.5, 0); \text{ Dashed Line, } (0, 0)\) 77

4.18 Probability Contents for \( n = 100 \) 81

4.19 Probability Contents for \( n = 200 \) 82

4.20 Probability Contents for \( n = 500 \) 83

4.21 Symmetric Difference of Optimization and KDE-Based Contours
with Truth 85

4.22 Symmetric Difference of Optimization and KDE-Based Contours
with Truth 86
4.23 Symmetric Difference of Optimization and KDE-Based Contours

with Truth ........................................ 87

4.24 Asymmetric Data .................................. 91

4.25 Asymmetric Data .................................. 93

4.26 Asymmetric Data .................................. 94

4.27 Asymmetric Data .................................. 95

4.28 Asymmetric Data .................................. 96

4.29 Asymmetric Data .................................. 97

4.30 Bimodal Data .................................... 100

4.31 Bimodal Data .................................... 101

4.32 Bimodal Data .................................... 102

4.33 Bimodal Data .................................... 104

4.34 Bimodal Data .................................... 105

4.35 Bimodal Data .................................... 106

4.36 Bimodal Data .................................... 107

C.1 End Contour with $\lambda_1 = 0.00$ and $\lambda_2 = 0.00 - 0.06$ ................. 120

C.2 End Contour with $\lambda_1 = 0.00$ and $\lambda_2 = 0.08 - 0.12$ ................. 121

C.3 End Contour with $\lambda_1 = 0.02$ and $\lambda_2 = 0.00 - 0.06$ ................. 122

C.4 End Contour with $\lambda_1 = 0.02$ and $\lambda_2 = 0.08 - 0.10$ ................. 123

C.5 End Contour with $\lambda_1 = 0.04$ and $\lambda_2 = 0.00 - 0.06$ ................. 124
C.6 End Contour with $\lambda_1 = 0.04$ and $\lambda_2 = 0.08$ ........................................... 125

C.7 End Contour with $\lambda_1 = 0.06$ and $\lambda_2 = 0.00 - 0.06$ ................................. 126

C.8 End Contour with $\lambda_1 = 0.08$ and $\lambda_2 = 0.00 - 0.06$ ................................. 127

C.9 End Contour with $\lambda_1 = 0.10$ and $\lambda_2 = 0.00 - 0.04$ ................................. 128

C.10 End Contour with $\lambda_1 = 0.12$ and $\lambda_2 = 0.00 - 0.02$ ................................. 129

C.11 End Contour with $\lambda_1 = 0.14$ and $\lambda_2 = 0.00$ ............................................ 130

C.12 Penalty on Perimeter= 0.0, Number of Vertices= 7—Solid Line,

25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%

Contour ......................................................... 132

C.13 Penalty on Perimeter= 0.0, Number of Vertices= 10—Solid Line,

25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%

Contour ......................................................... 133

C.14 Penalty on Perimeter= 0.0, Number of Vertices= 13—Solid Line,

25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%

Contour ......................................................... 134

C.15 Penalty on Perimeter= 0.0, Number of Vertices= 16—Solid Line,

25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%

Contour ......................................................... 135

C.16 Penalty on Perimeter= 0.02, Number of Vertices= 7—Solid Line,

25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%

Contour ......................................................... 136
C.17 Penalty on Perimeter= 0.02, Number of Vertices= 10—Solid Line,
   25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
   Contour ................................................................. 137

C.18 Penalty on Perimeter= 0.02, Number of Vertices= 13—Solid Line,
   25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
   Contour ................................................................. 138

C.19 Penalty on Perimeter= 0.02, Number of Vertices= 16—Solid Line,
   25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
   Contour ................................................................. 139

C.20 Penalty on Perimeter= 0.04, Number of Vertices= 7—Solid Line,
   25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
   Contour ................................................................. 140

C.21 Penalty on Perimeter= 0.04, Number of Vertices= 10—Solid Line,
   25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
   Contour ................................................................. 141

C.22 Penalty on Perimeter= 0.04, Number of Vertices= 13—Solid Line,
   25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
   Contour ................................................................. 142

C.23 Penalty on Perimeter= 0.04, Number of Vertices= 16—Solid Line,
   25% Contour; Dotted Line, 50% Contour; Dashed Line, 75%
   Contour ................................................................. 143
C.24 Number of Vertices= 7, Empirical Probability

Content= 0.25—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04 .................................. 144

C.25 Number of Vertices= 7, Empirical Probability Content= 0.5—Solid

Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04 .................. 145

C.26 Number of Vertices= 7, Empirical Probability

Content= 0.75—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 ......................................................... 146

C.27 Number of Vertices= 10, Empirical Probability

Content= 0.25—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 ......................................................... 147

C.28 Number of Vertices= 10, Empirical Probability

Content= 0.5—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 ......................................................... 148

C.29 Number of Vertices= 10, Empirical Probability

Content= 0.75—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 ......................................................... 149

C.30 Number of Vertices= 13, Empirical Probability

Content= 0.25—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 ......................................................... 150
C.31 Number of Vertices = 13, Empirical Probability

Content = 0.5—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 ................................................. 151

C.32 Number of Vertices = 13, Empirical Probability

Content = 0.75—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 .................................................. 152

C.33 Number of Vertices = 16, Empirical Probability

Content = 0.25—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 .................................................. 153

C.34 Number of Vertices = 16, Empirical Probability

Content = 0.5—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 .................................................. 154

C.35 Number of Vertices = 16, Empirical Probability

Content = 0.75—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed

Line, 0.04 .................................................. 155

C.36 Penalty on Perimeter = 0.0, Empirical Probability

Content = 0.25—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed

Line, 13; and Long-Dashed Line, 16 .................................. 156

C.37 Penalty on Perimeter = 0.0, Empirical Probability

Content = 0.5—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed

Line, 13; and Long-Dashed Line, 16 .................................. 157
C.38 Penalty on Perimeter = 0.0, Empirical Probability

Content = 0.75---Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed
Line, 13; and Long-Dashed Line, 16 ........................................ 158

C.39 Penalty on Perimeter = 0.02, Empirical Probability

Content = 0.25---Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed
Line, 13; and Long-Dashed Line, 16 ........................................ 159

C.40 Penalty on Perimeter = 0.02, Empirical Probability

Content = 0.25---Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed
Line, 13; and Long-Dashed Line, 16 ........................................ 160

C.41 Penalty on Perimeter = 0.02, Empirical Probability

Content = 0.75---Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed
Line, 13; and Long-Dashed Line, 16 ........................................ 161

C.42 Penalty on Perimeter = 0.04, Empirical Probability

Content = 0.25---Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed
Line, 13; and Long-Dashed Line, 16 ........................................ 162

C.43 Penalty on Perimeter = 0.04, Empirical Probability

Content = 0.5---Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed
Line, 13; and Long-Dashed Line, 16 ........................................ 163

C.44 Penalty on Perimeter = 0.04, Empirical Probability

Content = 0.75---Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed
Line, 13; and Long-Dashed Line, 16 ........................................ 164
C.45 Penalty on Area when Penalty on Perimeter=0.0—Solid Line,
\[ \nu = 7; \text{ Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16} \] 166

C.46 Penalty on Area when Penalty on Perimeter=0.02—Solid Line,
\[ \nu = 7; \text{ Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16} \] 167

C.47 Penalty on Area when Penalty on Perimeter=0.04—Solid Line,
\[ \nu = 7; \text{ Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16} \] 168

C.48 Penalty on Area when Number of Vertices=7—Solid Line,
\[ \lambda_2 = 0.0; \text{ Dotted Line, 0.02; Dashed Line, 0.04} \] 169

C.49 Penalty on Area when Number of Vertices=10—Solid Line,
\[ \lambda_2 = 0.0; \text{ Dotted Line, 0.02; Dashed Line, 0.04} \] 170

C.50 Penalty on Area when Number of Vertices=13—Solid Line,
\[ \lambda_2 = 0.0; \text{ Dotted Line, 0.02; Dashed Line, 0.04} \] 171

C.51 Penalty on Area when Number of Vertices=16—Solid Line,
\[ \lambda_2 = 0.0; \text{ Dotted Line, 0.02; Dashed Line, 0.04} \] 172

C.52 Seed = 17643 174

C.53 Seed = 29221 175

C.54 Seed = 15565 176

C.55 Seed = 20145 177
<p>| C.56 Seed = 6612 | 178 |
| C.57 Seed = 13663 | 179 |
| C.58 Seed = 19955 | 180 |
| C.59 Seed = 3062 | 181 |
| C.60 Seed = 64067 | 182 |
| C.61 Seed = 7548 | 183 |
| C.62 Seed = 7814 | 184 |
| C.63 Seed = 5114 | 185 |
| C.64 Seed = 14372 | 186 |
| C.65 Seed = 6058 | 187 |
| C.66 Seed = 12369 | 188 |
| C.67 Seed = 11927 | 189 |
| C.68 Seed = 21116 | 190 |
| C.69 Seed = 18203 | 191 |
| C.70 Seed = 169400 | 192 |
| C.71 Seed = 176540 | 193 |
| C.72 Seed = 22041 | 194 |
| C.73 Seed = 13749 | 195 |
| C.74 Seed = 210080 | 196 |
| C.75 Seed = 22219 | 197 |
| C.76 Seed = 74890 | 198 |</p>
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Area and Perimeter Penalties, End Contour Parameters</td>
<td>48</td>
</tr>
<tr>
<td>4.2 Number of Vertices, End Contour Parameters</td>
<td>49</td>
</tr>
<tr>
<td>4.3 Sample Size, Initial Radius, and End Contour Parameters</td>
<td>65</td>
</tr>
<tr>
<td>4.4 Samples Size, Initial Center, and End Contour Parameters</td>
<td>72</td>
</tr>
<tr>
<td>4.5 KDE- and Optimization-Based Contours—Mean Symmetric Differences</td>
<td>88</td>
</tr>
<tr>
<td>4.6 KDE- and Optimization-Based Contours—Median Symmetric Differences</td>
<td>89</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

While much work has been devoted to the non-parametric estimation of an unknown density function, comparatively little work has been focused on nonparametric estimation of the \textit{density contour}. The significance of such an estimate includes its usefulness in the inference about sets of minimum volume with given probability content, as in \textit{highest posterior density credible sets} in Bayesian theory.

Because the method of kernel estimates to pursue the first problem has received much attention, using the kernel estimate to estimate the density contour is a natural starting point. The boundary of the set of all points in $\mathcal{R}^2$ that have sample kernel density estimate at or above $\alpha$, a one-dimensional manifold, would be the non-parametric estimate of the density contour. Many statistical packages generate a sample non-parametric density contour estimate every time a contour plot is requested by the user. In the first portion of this paper, we investigate the properties of the one-dimensional density contour estimate derived from the kernel density estimate. In two dimensions, the properties of such an estimate are explored using Monte Carlo simulation.

The next direction we choose to explore is indicated due to the dual nature of a probability density contour itself. While the definition of such a contour is a set
of points at which the unknown density will have a constant value, this contour is also significant, as mentioned above, as a region in the plane of minimum area containing a given probability. So while basis of the estimation of a probability density contour on the evaluation of a kernel density estimate at some number of points is natural, it is equally natural to then turn to a method based on finding a "best" region with a desired probability content: "best" in our investigation being in terms of minimizing an objective function of the following form:

$$\hat{P}(A) + \lambda_1 \text{Area}(A) + \lambda_2 \text{Roughness}(A)$$

for $A$, a closed polygon of a given number of sides. The third term in the display above is necessary to avoid "pathological" contours. The measure of roughness used in these analyses is the perimeter of the polygon.

We examine the resulting contours when such a function is minimized for different values of the penalties, $\lambda_1$ and $\lambda_2$, on area and perimeter, respectively. We then compare the average behavior (for fixed values of the penalties) of the contour to a kernel-density-estimate based estimator in a Monte Carlo setting. Finally, we test this approach when the underlying, unknown distribution has interesting contour features, for example, bimodality or "banana-shaped" contours.
We show the following results:

- In one dimension, the kernel-based density contour estimate exists, is unique and consistent, and has an asymptotic distribution which we express given certain sufficient conditions.

- In two dimensions, the kernel-based density contour estimate, when parameterized in a certain fashion, shows characteristics of a Gaussian process.

- Direct estimation of a density contour is feasible using a Lagrange-multiplier-type method, wherein a candidate contour's probability content is determined by how much its area is penalized. The strength of this dependence (probability content on area penalty) is shown to increase with the sample size.

- Structure in the data may be shown in the optimization-based contour estimates through alteration of the penalties on area and perimeter.
Chapter 2

Basic Notions & Literature Review

The problem explored herein is estimation of the contours of an unknown density function, \( f(x) \), \( x \in \mathcal{R}^d \), \( d = 1, 2 \), given a random sample, \( X_1, \ldots, X_n \), from a distribution with that density. In spite of the vast work devoted to the closely related problem of estimation of the value of the density function itself, little work has been devoted to the estimation of the contours of the underlying probability density function, as is evidenced by the brevity of the literature cited below. The focus of our research is direct estimation of the density contours, i.e., the sets \( \mathcal{L}_\alpha = \{ x : f(x) = \alpha \} \) or the level sets, the sets \( \mathcal{C}_\alpha = \{ x : f(x) \geq \alpha \} \), for a given \( \alpha \). In one and two dimensions, we investigate the accuracy of using density contours from data-derived kernel estimates, \( \hat{\mathcal{L}}_\alpha = \{ x : \hat{f}(x) = \alpha \} \), to approximate the true, unknown contours. As an alternative to pointwise determination of these, in two dimensions, we define an objective function which takes in to consideration the probability content, area, and “roughness” (in the form of the perimeter) of the polygon. In order to derive density contours directly, we take as our estimator that contour which gives the least value for

\[-\hat{P}(A) + \lambda_1 \text{Area}(A) + \lambda_2 \text{Perimeter}(A)\]
for $A$, a closed polygon of a given number of sides, $\lambda_1 > 0$, the penalty on area, and $\lambda_2 > 0$, the penalty on perimeter.

One example application of such estimates would be construction of sets with minimum Lebesgue measure with a desired probability content, as in highest posterior density (HPD) credible sets in Bayesian theory [2]. These sets are of the form

$$C = \{ \theta \in \Theta : \pi(\theta|x) \geq k(\eta) \},$$

where $\Theta$ is the parameter space, $\pi(\theta|x)$ is the posterior distribution of the parameter $\theta$ given $x$, and $k(\eta)$ is the largest constant such that

$$P(C|x) \geq 1 - \eta.$$

The HPD region would correspond to the set $C_\alpha$ above, and the resulting size ($(1 - \eta)$ in the display above), or probability content, of the set could be estimated in one of two ways: either the proportion of the sample points $x_i$ with $\hat{f}_n(x_i) \geq \alpha$, or the result of the numerical integration of the kernel estimate itself over the HPD region, i.e.,

$$\int_{C_\alpha} \hat{f}_n.$$

The problem of directly estimating density contours has been addressed before, but not without some restriction as to the properties of the resulting estimate of the unknown contour. Sager in [12] proves convergence properties for estimators of the mode and isopleth (i.e., contour) of an unknown density function using a
method which iteratively finds the convex set of minimum volume containing a given proportion of the points from the previous step. A contour of a given size (probability content) is then estimated by the boundary of the subset corresponding to a given number of iterations, and Sager notes that the number of iterations is independent of $n$. The method assumes that the unknown distribution is strongly unimodal—in $\mathcal{R}^d$, that is, if $\theta$ is the mode, this implies that $f(x)$ decreases along any ray emanating from $\theta$—and that the level sets are convex. The author points out that while these are "potent" assumptions, that first, many distributions, in practice, seem to meet them approximately, and second, for local estimation (more appropriate in the case of estimating the mode than the isopleth), it is only necessary that the unimodality and convexity hold in some neighborhood of a local mode. These assumptions made, Sager proves the a.s. convergence of the sample isopleth to the true value. It should be emphasized that the paper is focused on properties of the sample isopleth once it is found. Sager points out that a method in $\mathcal{R}^1$ was given in Venter [18] or Robertson and Cryer [8], and that in higher dimensions, "the difficulty is real" to find the minimum volume convex set containing the requisite proportion of points.

Hartigan [3] took on the difficulty of constructing a convex set $S$ in two dimensions to approximate the true $\alpha$ contour of a density $f$, which he points out is the boundary of the set $S_0 = \{x|f(x) \geq \alpha\}$. The assumption made on $f$ in the asymptotics portion of this paper is that $f$ has continuous non-zero derivatives
near the convex contour $f = \alpha$. Hartigan continues by pointing out that that this is the largest set $S$ maximizing $(P - \lambda)S$, where

$$P(S) = \int_S f(x) \, dx \quad \text{and} \quad \lambda(S) = \alpha \int_S \, dx.$$  

In contrast to Sager’s paper, Hartigan’s includes an example of his algorithm finding a contour containing 50% of a sample for five different sample sizes and gives an order for the number of computations ($O(n^3)$). The limitation here is that the results are only for convex contours in $\mathcal{R}^2$.

Nolan [5] extends this approach to higher dimensions by further restricting the possible contours from convex sets to ellipsoids. Excess-mass ellipsoids (EME’s) are defined as the ellipsoids that maximize the difference between their probability content and a constant multiple of their volume, over all ellipsoids. Nolan cites several references for the construction of such sets, and points out that the computations required to find these in $\mathcal{R}^2$ are of $O(n^5)$, or possibly $O(n^6)$.
Chapter 3

Kernel-Based Density Contour Estimates

3.1 Kernel-Based Contour Estimation in One Dimension

3.1.1 Theory

In the one-dimensional case, the kernel density estimate is

\[ \hat{f}_n(x) = (nh_n)^{-1} \sum_{i=1}^{n} K \left( \frac{x - X_i}{h_n} \right). \]

It should be pointed out here that the bandwidth, \( h_n \), in addition to depending on the sample size, \( n \), will, in practice, depend on the data. In a setting where the true underlying distribution is unknown, the optimal bandwidth for estimating a particular contour will have to be approximated, based on the data gathered. To estimate probability "contours" for a given \( \alpha \) is to estimate \( x_\alpha \), where \( f(x_\alpha) = \alpha \).

To do this using the kernel density estimate, we need only find \( \hat{x}_{n\alpha} \) such that \( \hat{f}_n(\hat{x}_{n\alpha}) = \alpha \). The discussion in this subsection concerns the existence, consistency, uniqueness, and finally the distribution of \( \hat{x}_{n\alpha} \). Before getting to those results, however, we state some properties of the kernel estimate (all proofs relegated to chapter 6, starting on page 110). These properties will aid in understanding some results given without proof later on.

**Theorem 3.1** (Properties of \( \hat{f}_n \))
Given \( f, \hat{f}_n, \) and \( K \) satisfying \( \mathcal{F}_1 \) and \( \mathcal{K}_1 \), respectively, below:

1. \( f'' \) is absolutely continuous

\( \mathcal{F}_1 \) 
2. \( f''' \in L_2 \)

3. \( \exists f'' \)

1. \( K(x) \) is a uniformly continuous p.d.f. with finite support

\( \mathcal{K}_1 \) 
2. \( K(x) \) is symmetric about 0

3. \( K \in L_2 \)

Then \( h_n \to 0 \) and \( nh_n \to \infty \) as \( n \to \infty \) imply:

1. \( \text{E} \hat{f}_n(x) = f(x) + \frac{1}{2} \sigma^2_K h_n^2 f''(x) + O(h_n^4) \)

2. \( \text{Var} \hat{f}_n(x) = (nh_n)^{-1} f(x) R(K) - n^{-1} f^2(x) + O(h_n n^{-1}) \)

Where

\[ R(f) \overset{\text{def}}{=} \int f^2 \, dx \]

and

\[ \sigma^2_K \overset{\text{def}}{=} \int x^2 K(x) \, dx \]

The conclusions of Theorem 3.1 will help in understanding where the bias and variance terms come from when we give Parzen's distribution result below (Theorem 3.7). Before showing the a.s. existence of a unique and consistent estimate for the contour, we need stronger results regarding the weak and strong
consistency of the kernel estimate of a density and its derivatives from an article (of that name) by Silverman [16], where they are proved.

**Theorem 3.2 (Uniform Consistency of \( \hat{f}_n \))**

If the density function \( f \) is uniformly continuous on \( (-\infty, \infty) \), \( \hat{f}_n \) is as defined above with \( K \) satisfying

1. \( K \) is uniformly continuous and of bounded variation

2. \( \int |K(x)| \, dx < \infty \) and \( K(x) \to 0 \) as \( |x| \to \infty \)

\[ \mathcal{K}_2 \]

3. \( \int K(x) \, dx = 1 \)

4. \( \int |x \log|x||^{1/2} |dK(x)| < \infty, \)

and \( h_n \) satisfies

\[
h_n \to 0 \text{ and } nh_n (\log n)^{-1} \to \infty \text{ as } n \to \infty
\]

then

\[
\sup_x |\hat{f}_n(x) - f(x)| \xrightarrow{a.s.} 0 \text{ as } n \to \infty.
\]

No proof of this theorem (nor the next) is given here; however we point out that conditions \( \mathcal{K}_1 \) do imply conditions \( \mathcal{K}_2 \). Silverman in [16] notes that the conditions on \( h_n \) are sufficient conditions for the uniform consistency of \( \hat{f}_n \). In the following theorem from the same paper, the conditions on \( h_n \) are necessary as well. First, a definition (from Sen and Singer,[14]): for a function \( g : \mathcal{R} \to \mathcal{R} \), we define the
modulus of continuity, $w_g(\delta)$, as

$$w_g(\delta) = \sup_{s,t \in R} \{|f(t) - f(s)| : |s - t| < \delta\}.$$  

**Theorem 3.3 (Uniform Consistency of $\hat{f}_n'$)** Suppose the kernel function $K$ satisfies

1. $K$ is everywhere differentiable

2. For $j = 0, 1$, $K^{(j)}(x) \to 0$ as $x \to \infty$ and $\int |K^{(j)}(x)| dx < \infty$

3. $\int K(x) dx = 1$

4. $K'$ satisfies $K_2$ parts 1, 2, and 4

5. If we let $\gamma(u) = \{w_{K'}(u)\}^{1/2}$, then

$$\int [\log(1/u)]^{1/2} d\gamma(u) < \infty$$

6. The Fourier transform of $K$ is not identically one in any neighborhood of zero

and $f'$ is uniformly continuous. Then the conditions

$$h_n \to 0 \text{ and } nh_n^3 \{\log(1/h)\}^{-1} \to \infty \text{ as } n \to \infty$$

are necessary and sufficient for both

$$\sup |\hat{f}_n' - f'| \xrightarrow{P} 0 \text{ as } n \to \infty$$

and

$$\sup |\hat{f}_n' - f'| \xrightarrow{a.s.} 0 \text{ as } n \to \infty$$
Before we proceed to our first result directly involving the density contour, we need two results from real analysis. First, from Rudin (proof given in [11]).

**Theorem 3.4 (Existence of Limit Point)** If $E$ is an infinite subset of a compact set $K$, then $E$ has a limit point in $K$.

Second, from Royden [10] (as an exercise—proof given in chapter 6).

**Theorem 3.5 (Royden exercise)** $x_n \to x$ if and only if every subsequence of $(x_n)$ has in turn a subsequence that converges to $x$.

**Theorem 3.6 (One-Dimensional Density Contour Estimation)** Given $X_1, \ldots, X_n$ from a distribution with unknown p.d.f. $f$ satisfying $F_1$ above, $\alpha \in (0, \max \{f(x) : x \in \mathcal{R}\}), f(x_\alpha) = \alpha, f'(x_\alpha) \neq 0$. Then if $\hat{f}_n$ is defined as above with $K$ satisfying $K_3$, and if as $n \to \infty$, $h_n$ satisfies all of

$$h_n \to 0 \quad nh_n(\log n)^{-1} \to \infty \quad nh_n^3(\log(1/h))^{-1} \to \infty,$$

Then $\exists \epsilon > 0$ such that

$$P\{\exists N \ni \forall n > N \exists ! \hat{x}_{na} \in [x_\alpha - \epsilon, x_\alpha + \epsilon]$$

$$\exists \hat{f}_n(\hat{x}_{na}) = \alpha \text{ and } \hat{x}_{na} \to x_\alpha \} = 1$$

Theorem 3.6 implies that if there is a solution to $f(x) = \alpha$, it can almost surely be estimated uniquely and as precisely as desired. Our first step toward the goal of
deducing the distribution of any such \( \hat{x}_{n\alpha} \) is calling to mind a result from one of the hallmark papers in the topic of kernel density estimation, Parzen’s “On Estimation of a Probability Density Function and Mode,” [7], which we give without proof but point out that the conditions required here of \( K \) are implied by \( K_1 \).

**Theorem 3.7** *(Asymptotic Normality of \( \hat{f}_n(x_\alpha) \))*. Suppose \( \hat{f}_n \) is as defined above with \( X_1, \ldots, X_n \) from an unknown density \( f \). Suppose \( K \) is even and satisfies

1. \( \sup_{-\infty<y<\infty} |K(y)| < \infty \)
2. \( \int_{-\infty}^{\infty} |K(y)| dy < \infty \)
3. \( \lim_{y \to \infty} |yK(y)| = 0 \)
4. \( \int_{-\infty}^{\infty} K(y) dy = 1 \).

Then \( h_n \) satisfying \( h_n \to 0 \) and \( nh_n \to \infty \) as \( n \to \infty \) implies

\[
\frac{\hat{f}_n(x_\alpha) - (\alpha + \frac{1}{2}\sigma_{K}^2 h_n^2 f''(x_\alpha))}{(\alpha R(K)/nh_n)^{1/2}} \xrightarrow{D} N(0, 1)
\]

We now can take Theorem 3.7, the Mean Value Theorem, and Slutsky’s Theorem to arrive at the following result, ultimate in our discussion of the one-dimensional case.
Theorem 3.8 (Asymptotic Normality of \( \hat{x}_{n\alpha} \)) Given \( f \) satisfying \( \mathcal{F}_1 \), \( \hat{f} \) with \( K \) satisfying \( \mathcal{K}_3 \), \( h_n \) satisfying all of

\[
h_n \to 0 \quad nh_n(\log n)^{-1} \to \infty \quad nh_n^3(\log(1/h))^{-1} \to \infty,
\]

with

\[
\beta \overset{\text{def}}{=} -\frac{\sigma_K^2 f''(x_0)}{2f'(x_0)}
\]

\[
\tau \overset{\text{def}}{=} \frac{(\alpha R(K))^{1/2}}{|f'(x_0)|},
\]

then

\[
\frac{\hat{x}_{n\alpha} - [x_0 + \beta h_n^2]}{\tau(nh_n)^{-1/2}} \overset{d}{\rightarrow} N(0, 1)
\]

3.1.2 Discussion

With the theory for the basis of a contour estimate on a kernel density estimate in one-dimension given, some points need to be made.

- The conditions that \( h_n \to 0 \) and \( nh_n(\log n)^{-1} \to \infty \) as \( n \to \infty \) from Theorem 3.2 and the condition that \( nh_n\{\log(1/h)\}^{-1} \to \infty \) as \( n \to \infty \) from Theorem 3.3 are satisfied by \( h_n \sim n^{-a}, \forall 0 < a < 1/3 \), in general and for \( a = 1/5 \), the optimal bandwidth order for non-negative univariate kernel density estimators, in specific.

The optimal bandwidths for the pointwise mean-squared error (MSE) in the kernel density estimate, for the integrated MSE error in the kernel density
estimate, and for the error in estimating the $\alpha$-contour are given below. For the first two, optimal is in terms of minimizing asymptotic mean-squared error; because of the point made below, however, optimal in the case of the contour estimate has to be made in terms of minimizing the sum of the squared asymptotic bias and asymptotic variance of the approximating Gaussian distribution.

**Pointwise (at $x$) MSE for the Kernel Density Estimate**

$$h_p(x) = \left[ \frac{f(x)R(K)}{\sigma_k^2[f''(x)]^2} \right]^{1/5} n^{-1/5}$$

**Integrated MSE for the Kernel Density Estimate**

$$h_{IMSE} = \left[ \frac{R(K)}{\sigma_k^4 R(f'')} \right]^{1/5} n^{-1/5}$$

**$\alpha$-Contour Estimate**

$$h_{c}(\alpha) = \left[ \frac{\alpha R(K)}{\sigma_k^4[f''(x_\alpha)]^2} \right]^{1/5} n^{-1/5} \quad (3.1)$$

We note that the asymptotically optimal $h$ is the same for the $\alpha$-contour as the pointwise kernel density estimate—unsurprising, since the Taylor series result invoked to arrive at the distribution of the contour estimate relies on local linearity of the unknown density function.

- A more general result than Theorem 3.7 is found in a paper by Roussas [9]. In that paper, a degree of dependence is allowed among the $X_i$ with $X_i \in \mathcal{R}^d$. Restriction to one-dimensional, i.i.d. data in Roussas's paper, however, duplicates Parzen's result.
• We must be cautious in our interpretation of the $\beta$ and $\tau$ terms in Theorem 3.8. These cannot be thought of as moments, per se. As an example, consider density estimation in a case where the underlying distribution has infinite support: for any finite sample size, there will be a non-zero probability that there will be no sample solution $\hat{x}_{na}$—thereby making impossible definition of moments. These can be interpreted, however, as centering and normalizing constants in the context of the Central Limit Theorem, which is how they are used here.

3.1.3 Simulation

As with all asymptotic results, the theoretical results of subsection 3.1.1 need to be viewed critically, with the goal of understanding just how large $n$ needs to be in order for the results (especially the conclusions of Theorems 3.6 and 3.8) to hold. In this subsection we examine this question and consider the problem of normalizing the estimated $\hat{x}_{na}$'s when the optimal $h_n$ is unknown.

Before describing results, we give a brief description of notation: at the highest level, we refer to Monte Carlo experiments; within each experiment are a set number of trials; each trial consists of a fixed (within the experiment) number of samples, or i.i.d. replications. Using this terminology, what follow is a summary of the outcome of

2 “experiments” × 100 MC trials × (100 or 1000) N(0,1) samples.
To understand just how remote Asymptopia is, we did two experiments of 100 Monte Carlo trials. In both, we estimated the positive .05-contour from a standard normal distribution, where truth is \( x_{.05} = 2.038035 \). A Gaussian kernel was used. For the first experiment, each trial was of size \( n = 100 \) i.i.d. replications; the second was of size \( n = 1000 \). For each experiment, the same set of 100 random vectors was used twice—once with the optimal \( h \) for estimating the .05 contour (according to formula (3.1)) used for every trial, and once with the \( \hat{h}_n \) determined trial-to-trial using a biased cross-validation criterion. The intent of estimating the contour on the same set of vectors was to compare the ideal with the practical—in a real setting where estimation of the density contour is of interest, the \( h_n \) must be determined from the data.

For reference, here are \( \alpha, x_\alpha, f'(x_\alpha), f''(x_\alpha), \sigma_K, R(K) \), and the resulting \( \beta \), and \( \tau \) for all of these examples:

\[
\alpha = 0.05 \quad x_{.05} = 2.038035
\]
\[
f'(x_\alpha) = -0.1019018 \quad f''(x_\alpha) = .1576794
\]
\[
\sigma_K = 1 \quad R(K) = (4\pi)^{-1/2}
\]

Giving:

\[
\beta = 0.773683 \quad \tau = 1.165469
\]
The four plots which follow are QQ-plots output from S-PLUS. On each, the solid line is of slope 1, intercept 0, so a sample from a standard normal would be expected to "track" the line.

**Figure 3.1** QQ-Plot of $\hat{x}_n$'s from 100 Trials of 100 I.I.D. Replications Each and Fixed, Optimal Bandwidth

The first (Fig. 3.1) is from a run of 100 trials of size 100. In each of these trials, the fixed $h_C(\alpha) = .355$ was used in the kernel density estimate and the
standardization of the resulting $\hat{x}_{na}$'s (according to Theorem 3.8). The results do not appear significantly different from a standard normal.

Figure 3.2  QQ-Plot of $\hat{x}_{na}$'s from 100 Trials of 100 I.I.D. Replications Each and BCV-Determined Bandwidth

In our second example (Fig. 3.2), where the bandwidth was determined trial-to-trial, each $\hat{h}_n$ was used both as the bandwidth in the kernel density estimate for
that trial and in the standardization of the corresponding $\hat{x}_{n\alpha}$. Normality again appears to be a reasonable hypothesis.

**Figure 3.3** QQ-Plot of $\hat{x}_{n\alpha}$'s from 100 Trials of 1000 I.I.D. Replications Each and Fixed, Optimal Bandwidth

In our third example (Fig. 3.3), we increased the number of i.i.d. replications per trial to 1000 and used the fixed $h_C(\alpha) = 0.224$. The agreement with the stan-
dard normal is better than with \( n = 100 \), although the upper tail looks (perhaps insignificantly) heavy.

![QQ-Plot of \( \hat{x}_{na} \)'s from 100 Trials of 1000 I.I.D. Replications Each and BCV-Determined Bandwidth](image)

**Figure 3.4** QQ-Plot of \( \hat{x}_{na} \)'s from 100 Trials of 1000 I.I.D. Replications Each and BCV-Determined Bandwidth

In the last example in the one-dimensional case (Fig. 3.4), the biased cross-validation criterion was again used to determine \( \hat{h}_n \) for each trial. As in the case with \( n = 100 \) samples per trial, this \( \hat{h}_n \) was then used both for the kernel density
estimate and the standardization of the \( \hat{x}_{n\alpha} \)'s. The similarity to the QQ-plot using the fixed bandwidth (Fig. 3.3) indicates that the upper tail behavior is an artifact of the specific trials, rather than the choice of bandwidth. In any case, normality does not appear to be an unreasonable conclusion.

We close our discussion of the one-dimensional case by noting that in all four experiments—100 versus 1000 Monte Carlo i.i.d. replications, both the optimal and data-derived kernel bandwidth—normality is strongly substantiated for the appropriately normalized value of the density contour estimate. The only additional observation is that perhaps, in the case of 100 i.i.d. replications (Figures 3.1 and 3.2,) the sample size is insufficient to warrant use of Theorem 3.8, at least without fourth-order terms included in the centering constant \( \beta \), since the points on the QQ-plots indicate a slightly negative mean. Examination of the 1000 i.i.d. replication cases (Figures 3.3 and 3.4) shows no such consistent (however slight) departure from the standard zero mean.
3.2 Kernel-Based Contour Estimation in Higher Dimensions

3.2.1 Theory

In general, a multivariate kernel density estimate is of the form

\[ \hat{f}_n(x) = \frac{1}{n |H|} \sum_{i=1}^{n} K(H^{-1}(x - X_i)), \]

where the matrix \( H \) would allow the kernel function \( K \) to be evaluated in directions not necessarily along the coordinate axes. For our use we will restrict ourselves to product kernels in two dimensions, where the same (univariate) kernel is used in each dimension, but different bandwidths can be chosen for each. The formula for this is:

\[ \hat{f}_n(x) = \frac{1}{nh_1h_2} \sum_{i=1}^{n} \left\{ \prod_{j=1}^{2} K \left( \frac{x_j - x_{ij}}{h_j} \right) \right\}, \]

where \( x_{ij} \) is the \( j^{th} \) component of the observed \( X_i \) and \( x_j \) is the \( j^{th} \) component of the point \( x = (x, y) \) where the kernel density estimate is being calculated.

3.2.2 Simulation

The hypothesis we explore in the two-dimensional case is that the sample \( \alpha \)-contour will be the realization of a Gaussian process with a variance/covariance structure that we can calculate. In our simulation, we use theory from Ash and Gardner [1] that if the Karhunen-Loève expansion of a Gaussian process is calculated, the coefficients will be independent and normally distributed. For an explanation of this expansion, see Appendix B, from [1, §1.4].
In our examples, we use the following theorem (proof in 6):

**Theorem 3.9 (Basis Functions for a Rotationally Invariant Process)**

For a random \( L_2 \) process \( \{X(t), t \in [0, 2\pi]\} \), we say \( X(t) \) is rotationally invariant if \( E[X(t)] = c \ \forall \ t \in [0, 2\pi] \) (the constant \( c \) may be taken WLOG to be 0) and \( \text{Cov}[X(s), X(t)] = E[X(s)X(t)] = \kappa(s \ominus t) \) for some function \( \kappa() \) of one variable, where \( \ominus \) denotes subtraction modulo \( 2\pi \). If \( X(t) \) is rotationally invariant, then complex exponentials form the orthonormal basis for the Karhunen-Loève expansion of \( X(t) \).

The application of this theorem is immediate: the complex exponentials are the basis functions used in the Fourier transform, so if we would like the Karhunen-Loève coefficients for a rotationally invariant process, we may use the discrete Fourier transform in S-PLUS to approximate these coefficients. Note that for the second example, the Karhunen-Loève expansion is *not* a Fourier series, and the DFT from S-PLUS merely gives an approximation to the Fourier coefficients.

In both the examples which follow, the \( \alpha = .03 \)-contour in \( R^2 \) was estimated using \( n = 200 \) i.i.d. replications per Monte Carlo trial for 100 trials. For each, the contour was estimated along 128 equally-spaced rays around the circle—the value returned for each ray was the distance from the origin along that ray at which the kernel density estimate dropped below 0.03. Once a sample contour value was determined along each ray, the discrete Fourier transform (DFT) was taken of the
vector of 128 contour values. The analyses which follow are examinations of the means, variance, and QQ-plots for corresponding DFT coefficients across trials.

In the first example, we sampled from a Normal distribution with mean vector and variance/covariance matrix

$$\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

respectively. This gives a circular contour of radius approximately 1.826. We expect this to be reflected in the DFT coefficients by a large first real (DC) coefficient and the later coefficients to drop off in magnitude very quickly. This is reflected in Figure 3.5, where the first coefficient was not included, because at 241.6128, the others would have been indistinguishable from zero. Note that the means generally taper to zero (in fact, the means for coefficients 41-128 are not even included, as they are all practically zero). According to the Karhunen-Loève theorem, the means should be exactly zero, except for the DC component.

In plotting the variances, the $y$-axis is on a logarithmic scale (Fig. 3.6). If, as we hypothesize, the sample contour can be described as the realization of a Gaussian process, we would expect the weight given higher frequency terms in the DFT to diminish in a fashion that would make the coefficients' variances summable. The plot does not appear to refute this hypothesis. (Note that the variances given include that for the first coefficient.) Indeed the plot suggests an exponential rate of decrease.
Figure 3.5  Mean of DFT Coefficients 2-40 for Circular Contour
Figure 3.6  Variance of DFT Coefficients 1-40 for Circular Contour
Finally, in the first true test of whether the sample contour can be considered as a realization of a Gaussian process, a QQ-plot was done for each of the first nine DFT coefficients. Those plots are in Figure 3.7. In each, the coefficients have been centered by their mean and standardized by their standard deviation, so all should be approximately Normal(0, 1) (line of slope 1, intercept 0 indicated on each plot). The agreement with the ideal is strong.

The same analyses undertaken above were repeated in the case of a mixture density; each observation taken with probability = .5 from one of two normal distributions with parameters:

\[
\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}
\]

or

\[
\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}
\]

We refer to this simulation as the "Diamond Contour" problem—examination of the true \( \alpha = .03 \) contour in Figure 3.8 reveals why.

In examining the means of the DFT coefficients in this case, in addition to a strong DC component as in the first example, we expect a large real coefficient for the fourth Fourier frequency. This is seen to be the case in Figure 3.9. The value of the omitted first coefficient is 237.1485.

The variances of the DFT coefficients are given in Figure 3.10. As before, the plot is given with a log-log scale.
Figure 3.7  QQ-Plots for Standardized DFT
Coefficients 1-9 for Circular Contour
Figure 3.8  True ($\alpha = 0.03$) Diamond Contour
Figure 3.9  Mean of DFT Coefficients 2-40 for Diamond Contour
Figure 3.10  Variance of DFT Coefficients 1-40 Diamond Contour
As was the case in the first example, the coefficients, when appropriately stan-
dardized, seem to be approximately standard normal. Examination of the QQ-
plots in Figure 3.11 bears this out.

Figure 3.11  QQ-Plots for Standardized DFT
Coefficients 1-9 for Diamond Contour
Chapter 4

Polygon Optimization-Based Contour Estimation

4.1 Description of Optimization

Faced with the prospect of estimating an unknown probability contour in a pointwise fashion using kernel-density based estimators, we turn our attention to a generalization of the method described by Hartigan in [3], a *JASA* article, "Estimation of a Convex Density Contour in Two Dimensions." Basing his search for a contour on its characterization as the boundary of a region of given probability content with minimum area, he sets out to find an estimate of \( S \), say \( S_n \), a two-dimensional region which maximizes \( P(S) - \text{Area}(S) \), taking as his contour estimate \( \delta(S_n) \), boundary of \( S_n \). The problem with this method is that applying this to the empirical distribution of a sample \( X_1, \ldots, X_n \) gives, as its optimal \( S_n \), the union of the sample points. To avoid this pathological case, Hartigan restricted his search for density contours to the boundaries of convex sets with data points as the vertices of the set (convex polygons).

Like Hartigan, we will restrict our attention to the two-dimensional case and estimation using polygonal regions; unfortunately, also like Hartigan, we must overcome the tendency of an optimization of this sort toward the union of the sample points. Rather than restrict our attention to convex polygons, however,
we do this in the following fashion: we introduce a penalty for the "roughness" of
the estimated contour. Informally, we search for the set \( S_n \) maximizing \( \hat{P}_n(S) - \lambda_1 \text{Area}(S) - \lambda_2 \text{"Roughness"}(S) \), where the roughness measure for the set \( S \) is,
for example, its perimeter and \( \lambda_1, \lambda_2 \) are tuning parameters: \( \lambda_1 \) is determined to
obtain the desired probability content. This maximization is the "optimization"
referred to in the title of this section. Three things of note: first, the operator
notation employed by Hartigan has been replaced by a Lagrange-multiplier nota-
tion, wherein a candidate contour's probability content is penalized by its area and
roughness; second, the \( n \) subscript in all of these displays refers to the number of
observations on which the contour estimate is based; and finally, the \( \hat{P}_n \) will soon
be replaced by \( \hat{P}_n \), a "smoothed" empirical probability.

4.2 Algorithm

If we define

\[
\Omega(S) = \hat{P}_n(S) - \lambda_1 \text{Area}(S) - \lambda_2 \text{Perimeter}(S)
\]  

(4.1)

as our objective function evaluated for some sample size \( n \), our goal is to move the
vertices of \( S \) in the direction that would result in the greatest improvement of \( \Omega \).

In the following sections, we will first discuss computing the values of \( \hat{P}_n(S) \) and
Area\( (S) \) (recognizing the ease of computation of the perimeter,\) and then discuss
the efforts necessary to optimize their linear combination. We will refer to the
algorithm to find contour estimates via optimization of polygons as \textit{polyopt}. 

Before we discuss evaluation of polygons, we define them: from [6], a *polygon* is the region of a plane bounded by a finite collection of line segments forming a simple closed curve. Different mathematical disciplines define "simple closed curve" in different fashions, so we borrow further from O'Rourke: if we call $\nu$ the number of vertices, let $(x_1, y_1), \ldots, (x_\nu, y_\nu)$ be the vertices in *counter-clockwise* order starting from any initial vertex and if we call $e_1 = (x_1, y_1)(x_2, y_2)$ the first segment, proceeding around all the vertices until we reach $e_\nu = (x_\nu, y_\nu)(x_1, y_1)$, then the segments $\{e_i\}$ bound a polygon iff:

1. The intersection of each pair of adjacent segments is the single point shared between them: the common vertex, $(x_{i+1}, y_{i+1})$ for $e_i \cap e_{i+1}$ where $i = 1, \ldots, \nu - 1$ and $(x_1, y_1)$ for $e_\nu \cap e_1$. (Note: O’Rourke in [6] avoids these dual cases by making all index arithmetic throughout the text mod $\nu$.)

2. Nonadjacent segments do not intersect: $e_i \cap e_j = \emptyset$ when $j \neq i + 1 \mod \nu$.

From an implementation standpoint, the first of these conditions is satisfied simply by defining the segments as running from one two-dimensional point to the next. The second requires a simple but important routine that merely checks whether any sides cross one another; if so, adding a penalty to the objective function, thereby deterring *polyopt* from pursuing a solution any further in that direction. This solution wasn’t reached without consequences—c.f. section 4.3.1.
4.2.1 Computation of Probability Content and Area

With polygons defined and a check in place to assure their validity, we now turn our attention to the apparently thorny task of figuring out the empirical probability content of a given candidate contour, i.e., what proportion of a sample lies interior to it. It was with an eye on differentiating between interior and exterior that the convention above, that the vertices of the polygon in numerical order trace it's boundary in a counter-clockwise direction, was given. Once this is in place, we can determine whether a single point is interior to a polygon by doing the following: draw a ray from the point in (any, but we chose) the $+x$ direction and count the number of intersections between the ray and the segments of the polygon. If the number of crossings is even, the point is exterior to the polygon; otherwise, it is interior. With that simple geometric picture in mind, what remains is noticing that if we shift the point in question to the origin (and the vertices of the polygon by the same amount,) counting the number of intersections amounts to counting the number of segments which cross the positive $x-$axis. While the procedure in [6] actually does this shift, because we need to check every point in our data set, it was a simple modification to leave the point and the vertices where they are found.

For the second term in 4.1, the area of the polygon, we again turn to [6], where we are reminded of a result from linear algebra: that the magnitude of the cross product of two vectors is the area of the parallelogram they determine. The determinant form of this result (found in [6, §1.4.3, Lemma 1.4.1]) follows: if a
triangle $T$ has as its corners $(x_1, y_1), (x_2, y_2),$ and $(x_3, y_3),$ then

$$2\text{Area}(T) = \left| \begin{array}{ccc} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{array} \right| = x_1y_2 - y_1x_2 + y_1x_3 - x_1y_3 + x_2y_3 - x_3y_2$$

The astute reader will quickly note that the above quantity may be negative, (apparently) a drawback when considering its use as area. However, we point out quickly that that is only the case when $(x_1, y_1), (x_2, y_2),$ and $(x_3, y_3)$ are given in clockwise order—a fact that is, in fact, necessary for the next result (proof in [6]):

**Theorem 4.1 (Area of Polygon)**

Let a polygon $S$ have vertices $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ labeled counterclockwise; then

$$2\text{Area}(S) = \sum_{i=1}^{n} (x_iy_{i+1} - y_ix_{i+1})$$

The proof uses the determinant form above, with the initial vertex, $(x_1, y_1),$ used as one vertex in each of the triangles formed when using every adjacent pair of vertices for the other two—cancellation of like terms (with opposite signs) gives the simple form. That this works for convex as well as non-convex polygons is due to the “negative charge” given clockwise-numbered triangles.

Of note, especially in light of the motivation of the use of perimeter as a measure of roughness which follows, is that O’Rourke points out that
This theorem can be viewed as a discrete version of Green's theorem, which relates an integral around the boundary of a region, with an integral over the interior of the region: \( \int_{\partial \mathcal{P}} \omega = \int_{\mathcal{P}} d\omega \) where \( \omega \) is a "1-form" ... 

This is of note because of the characterization of perimeter, not as the sum of distance between vertices (as we do in the case of polygons,) but rather as is done with smooth curves: if we have a smooth curve, \( S = \{(x(t), y(t) : t \in [a, b]\}, \) then the arc length of \( S \), \( L(S) = \int_{a}^{b} \left\{ \left( \frac{dx}{dt} \right)^2 + \left( \frac{dy}{dt} \right)^2 \right\}^{1/2} dt. \) This gives an explicit link between perimeter and a derivative, albeit only in the case of smooth curves.

4.2.2 Non-Smoothness of the Objective Function

If we consider the summands in 4.1, two of the three are continuous with respect to movement of the vertices—\( \lambda_1 \text{Area}(S) \), a constant multiple of the area of the candidate polygon, and \( \lambda_2 \text{Perimeter}(S) \), a constant multiple of the perimeter. The summand that presents us with difficulty, optimization-wise, is \( \hat{P}(S) \), the empirical probability content of the polygon.

We would like to use a "steepest ascent algorithm," evaluating the objective function in small neighborhoods around the coordinates of the vertices of the current polygon, and then doing a linear fit (in the 2 * (no. of vertices)-dimensional space) of the objective function to the amounts the coordinates have been moved. This would give us, in effect, an estimate of the "gradient" at the current poly-
gon, and we proceed in the direction of the "gradient" as long as it results in an improvement of our objective function.

There are two major obstacles to the practical application of this procedure—first, the quick increase in number of variables as the number of vertices increases, and the high degree of dependence between the \( x \)-coordinate and \( y \)-coordinate of a given vertex and the dependence between these and the corresponding coordinates of the vertices which precede and follow the given vertex. This brings up the second obstacle—the discontinuity of the empirical probability content of the polygon as a function of the \((x, y)\) coordinates of the vertices. Movement of a vertex by a small amount can result in a discontinuous jump in the number of points contained in the polygon. These factors combined are worse than the sum of their parts: this change (a point leaving the polygon or a point entering the polygon) is not necessarily in a neighborhood of the vertex being moved. Because moving a vertex moves the segments connecting it to adjacent vertices, the polygon may gain or lose data points anywhere in a region tapering to a point at the next vertex.

These factors led to predictably bad results: the gradient-based search for an optimal polygon, relying on the independence of the input variables and lost in the \( n = \{\# \text{ data points}\} \) local maxima, arrived at inconsistent results. An increase in the penalties on area and perimeter sometimes resulted in an increase in their values; a small change in the initial guess resulted in a dramatic change in the end contour estimate. This problem of dimensionality and interdependence
of variables was (partially) solved by using a non-gradient optimization program called "pdsopt." This program is an implementation of a direct search algorithm developed (as was pdsopt) by Virginia Torczon [17]. It should be mentioned at this point, that while it has no bearing on the end solution, functionally Torczon’s routine returns the values for the input variables that result in a minimum for the objective being considered, so as experimental values are being considered below in tables or graphs, the reader should keep in mind that the negative of the function in 4.1 is being shown.

With the objective function adapted for calling by it, pdsopt performed better than the gradient-based procedure, because it didn’t rely on the independence of the input variables, but was still arriving at inconsistent results. The problem was in the second major obstacle mentioned above—that of the discontinuity of the empirical probability content of the contour estimate. Torczon’s routine was still getting lost in the vast number of local minima which arose as movement of the vertices would cause the edge of the contour estimate to approach (from the outside) data points, resulting in some amount of probability for the given area and perimeter, and then to cross over these data points, giving practically the same area and perimeter, but measurably less probability.

To combat this, we altered how empirical probability content of a contour was measured. Rather than every observation being considered “in” or “out,” a 1/n or 0 as probability is concerned, its proximity to the boundary of the contour
estimate (whether an edge or a vertex) was calculated and used in weighing how much that particular observation should be “in” or “out.” If a point’s proximity to the boundary was greater than some specified value and the point was interior to the contour estimate, it was counted as fully “in” (with accompanying probability 1/n); correspondingly distant on the exterior of the contour estimate and it was counted as fully “out” (with empirical probability 0.) A point lying exactly on the boundary counted for probability 0.5/n. This “fuzzy probability” is quantified according to the function:

\[
\tilde{P}(d) = \begin{cases} 
0 & d < -h_b \\
-1/4(d/h_b)^3 + 3/(4h_b) + 1/2 & -h_b \leq d \leq h_b \\
1 & h_b < d 
\end{cases}
\]

where \(d\) is the shortest distance from a point in question to the contour and the positive direction corresponds to the interior of the polygon. This is pictured below (Figure 4.1.)

This specified value to which the distances of the observations from the boundary of the contour estimate were compared (0.1 in Figure 4.1) to determine how much of them lie within the contour is analogous to bandwidth in kernel density estimation, and with that in mind, the value used will be designated as \(h_b\). No concerted effort was made in the course of this research to determine “optimum” values for this “bandwidth,” and it was only altered in a relatively few cases, all within the asymmetric/bimodal data section below (4.3.5).
Figure 4.1 Proportion of Point Deemed Interior to Polygon
4.3 Simulation Results

Because of the complexity of both the theoretic solution of the problem of finding an "optimal" polygon and the measures described above to computationally solve it, we rely, as we did in exploring the properties of the two-dimensional kernel-based contour estimate, on Monte Carlo simulation to assess the reasonability of our procedure for estimation of a density contour via optimization of a polygon.

For these simulation runs, the definition of the objective function in 4.1 is altered slightly, reflecting the use of "smoothed" probability (\(\hat{P}_n\)) and that \textit{pdsopt} searches for a minimum rather than a maximum:

\[
\hat{\Omega}(S) = -\hat{P}_n(S) + \lambda_1 \text{Area}(S) + \lambda_2 \text{Perimeter}(S)
\]  

(4.2)

The values of \(\lambda_1\) and \(\lambda_2\) were fixed, treated as input parameters for \textit{polyopt} (within a given run.)

The following issues are explored in the following sections:

- How do the penalties on area and perimeter and the number of vertices making up the polygon affect the final contour estimate's shape, probability content, area, perimeter, and the linear combination of these last three, the objective function?

- Using our general understanding of the behavior of the optimization routine, can we find a penalty on area, \(\lambda_1\), that will coerce \textit{polyopt} to arrive at a contour estimate of a given probability content? How does this depend on
the penalty on perimeter? Does the number of vertices change these relationships?

- How does the initial "size" or center of the contour estimate affect the end result? Does this depend on the sample size?

- How do estimates derived in this fashion compare to those from some "standard" method, for example, those derived simply by finding a kernel-based contour estimate with approximately the same probability content?

- Since an area of focus of research in this area has always been investigation of modality (or shape) of unknown an distribution, how does this method perform when data is asymmetrical or bimodal?

### 4.3.1 Effect of Penalties, $\lambda_1$, and Number of Vertices

If viewed practically, we expect that if polyopt is working as expected, then increasing the penalties on area ($\lambda_1$) or perimeter ($\lambda_2$), either individually or in tandem, would result in a smaller end contour estimate—that is, empirical probability content is sacrificed in order to reduce the area and/or perimeter. That hypothesis was tested by repeatedly running polyopt to completion with the same set of $n = 200$ i.i.d. replications of Gaussian data with mean $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and variance-covariance matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, keeping the number of sides and initial guess constant as a regular 10-gon of radius (distance from center $(0,0)$ to any vertex) $1.3$, changing only the values
of $\lambda_1$ and $\lambda_2$, and examining the end contour's shape, probability content, area, perimeter, and objective function value.

The only purpose served in examining the shape of the final contour would be some subjective assessment of "reasonability," although visual examination would reveal whether a purported density contour candidate had missed the mark entirely, e.g., by encircling a region of positive area with no points therein. Happily, that didn't happen, and so the graphs of the data with end contours for the given $(\lambda_1, \lambda_2)$'s have been relegated to Appendix C, section C.1.

Of note in the graphs, by virtue of supporting intuition as to how the end contour should behave with changing penalties on area and perimeter, are three phenomena: first, the monotone decrease in probability content as either or both penalties are increased. Second, as can be seen especially in Figures C.1, C.3 and C.5 (upper left-hand graph in each,) in situations where there is no penalty on perimeter, the end contour contains concave vertices, i.e., vertices where the measure of the interior angle is greater than that of the exterior angle—evidence that $polyopt$ is reducing area, which is being penalized, at a cost of perimeter, which is not. These figures also make good arguments for use of perimeter as a measure of "roughness." Third and finally, in the cases $\lambda_1 = 0$ and $\lambda_2 = 0.02, \ldots, 0.10$ in Figures C.1 and C.2, note in the corresponding case where there is no penalty on area and a non-zero penalty on perimeter, the end contour is very regular.
Numerical evidence that the "intuition supporting" results mentioned above are provided in the table which follows. Table 4.1, shows the penalty on area, penalty on perimeter, the end contour's probability content, area, perimeter, and the resulting objective function value, respectively. While the lower limit of the penalties, 0, is natural, the upper limit for each depended on the value of the other, and basically was found by incrementing them in units of 0.02 until a pathological end contour (i.e., a contour of little or no area) was the result. Note that in those cases, positive probability content may result, due to the fashion in which points are "phased" in and out.

As mentioned at the start of this section, supporting figures are found (in the form of all the end contours for all the $(\lambda_1, \lambda_2)$ ordered pairs) in appendix C, section C.1.

As above, to examine the effect of changing the number of vertices of the estimating polygon, we use the same set of $n = 200$ i.i.d. replications for all trials, a starting radius of 1.3, and center $(0, 0)$. In this section, however, the penalties on area and perimeter remained fixed at $(\lambda_1, \lambda_2) = (0.02, 0.04)$ while the number of vertices were set at $3, 4, \ldots, 20$. Although all the same parameters were measured as in the previous section (Table 4.2,) only the probability content and objective function are considered here graphically.

As can be seen in the table above and in the plot (Figure 4.2) below, the probability content is variable at a small number of vertices, and less so as the
<table>
<thead>
<tr>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>Probability</th>
<th>Area</th>
<th>Perimeter</th>
<th>Obj. Fcn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>145.2058</td>
<td>47.0983</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0.02</td>
<td>0.9587</td>
<td>15.083</td>
<td>14.3148</td>
<td>-0.6724</td>
</tr>
<tr>
<td>0</td>
<td>0.04</td>
<td>0.8956</td>
<td>11.4152</td>
<td>12.3123</td>
<td>-0.4031</td>
</tr>
<tr>
<td>0</td>
<td>0.06</td>
<td>0.7956</td>
<td>7.8699</td>
<td>10.2559</td>
<td>-0.1803</td>
</tr>
<tr>
<td>0</td>
<td>0.08</td>
<td>0.7055</td>
<td>6.1824</td>
<td>9.0762</td>
<td>0.0206</td>
</tr>
<tr>
<td>0</td>
<td>0.1</td>
<td>0.3689</td>
<td>2.362</td>
<td>5.6383</td>
<td>0.1949</td>
</tr>
<tr>
<td>0</td>
<td>0.12</td>
<td>0.0058</td>
<td>0.0078</td>
<td>0.7326</td>
<td>0.0821</td>
</tr>
<tr>
<td>0.02</td>
<td>0</td>
<td>0.9145</td>
<td>11.8218</td>
<td>13.6012</td>
<td>-0.6781</td>
</tr>
<tr>
<td>0.02</td>
<td>0.02</td>
<td>0.8629</td>
<td>9.6186</td>
<td>11.7836</td>
<td>-0.4349</td>
</tr>
<tr>
<td>0.02</td>
<td>0.04</td>
<td>0.74</td>
<td>6.7018</td>
<td>9.5517</td>
<td>-0.2239</td>
</tr>
<tr>
<td>0.02</td>
<td>0.06</td>
<td>0.6241</td>
<td>5.0976</td>
<td>8.262</td>
<td>-0.0264</td>
</tr>
<tr>
<td>0.02</td>
<td>0.08</td>
<td>0.3764</td>
<td>2.3809</td>
<td>5.7025</td>
<td>0.1275</td>
</tr>
<tr>
<td>0.02</td>
<td>0.1</td>
<td>0.0058</td>
<td>0.0078</td>
<td>0.7326</td>
<td>0.0676</td>
</tr>
<tr>
<td>0.04</td>
<td>0</td>
<td>0.7983</td>
<td>7.8716</td>
<td>11.3586</td>
<td>-0.4835</td>
</tr>
<tr>
<td>0.04</td>
<td>0.02</td>
<td>0.7092</td>
<td>6.1044</td>
<td>9.3187</td>
<td>-0.2786</td>
</tr>
<tr>
<td>0.04</td>
<td>0.04</td>
<td>0.6826</td>
<td>5.7927</td>
<td>8.8041</td>
<td>-0.0987</td>
</tr>
<tr>
<td>0.04</td>
<td>0.06</td>
<td>0.4852</td>
<td>3.4811</td>
<td>6.825</td>
<td>0.0636</td>
</tr>
<tr>
<td>0.04</td>
<td>0.08</td>
<td>0.0042</td>
<td>0.0024</td>
<td>0.6641</td>
<td>0.049</td>
</tr>
<tr>
<td>0.06</td>
<td>0</td>
<td>0.6859</td>
<td>5.5509</td>
<td>10.1665</td>
<td>-0.3528</td>
</tr>
<tr>
<td>0.06</td>
<td>0.02</td>
<td>0.643</td>
<td>5.1297</td>
<td>8.8477</td>
<td>-0.1582</td>
</tr>
<tr>
<td>0.06</td>
<td>0.04</td>
<td>0.4585</td>
<td>3.1975</td>
<td>6.6332</td>
<td>-0.0014</td>
</tr>
<tr>
<td>0.06</td>
<td>0.06</td>
<td>0.0087</td>
<td>0.0137</td>
<td>0.6594</td>
<td>0.0317</td>
</tr>
<tr>
<td>0.08</td>
<td>0</td>
<td>0.5996</td>
<td>4.3699</td>
<td>9.7416</td>
<td>-0.25</td>
</tr>
<tr>
<td>0.08</td>
<td>0.02</td>
<td>0.3827</td>
<td>2.3239</td>
<td>5.884</td>
<td>-0.0791</td>
</tr>
<tr>
<td>0.08</td>
<td>0.04</td>
<td>0.2932</td>
<td>1.6296</td>
<td>4.8117</td>
<td>0.0296</td>
</tr>
<tr>
<td>0.08</td>
<td>0.06</td>
<td>0.0056</td>
<td>0.0071</td>
<td>0.7455</td>
<td>0.0397</td>
</tr>
<tr>
<td>0.1</td>
<td>0</td>
<td>0.4059</td>
<td>2.3638</td>
<td>8.4016</td>
<td>-0.1695</td>
</tr>
<tr>
<td>0.1</td>
<td>0.02</td>
<td>0.2519</td>
<td>1.2424</td>
<td>4.8276</td>
<td>-0.0311</td>
</tr>
<tr>
<td>0.1</td>
<td>0.04</td>
<td>0.0184</td>
<td>0.0415</td>
<td>1.3657</td>
<td>0.0403</td>
</tr>
<tr>
<td>0.12</td>
<td>0</td>
<td>0.394</td>
<td>2.3158</td>
<td>7.5174</td>
<td>-0.1161</td>
</tr>
<tr>
<td>0.12</td>
<td>0.02</td>
<td>0.0367</td>
<td>0.1214</td>
<td>1.7142</td>
<td>0.0121</td>
</tr>
<tr>
<td>0.14</td>
<td>0</td>
<td>0.1219</td>
<td>0.477</td>
<td>6.475</td>
<td>-0.0552</td>
</tr>
</tbody>
</table>

Table 4.1  Area and Perimeter Penalties, End Contour Parameters
<table>
<thead>
<tr>
<th>No. Vertices</th>
<th>Probability</th>
<th>Area</th>
<th>Perimeter</th>
<th>Obj. Fcn</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.6273</td>
<td>6.5444</td>
<td>11.704</td>
<td>-0.0282</td>
</tr>
<tr>
<td>4</td>
<td>0.6338</td>
<td>5.4329</td>
<td>9.3618</td>
<td>-0.1507</td>
</tr>
<tr>
<td>5</td>
<td>0.8051</td>
<td>8.5898</td>
<td>11.3162</td>
<td>-0.1806</td>
</tr>
<tr>
<td>6</td>
<td>0.7617</td>
<td>7.0522</td>
<td>10.1977</td>
<td>-0.2128</td>
</tr>
<tr>
<td>7</td>
<td>0.7283</td>
<td>6.6959</td>
<td>9.5697</td>
<td>-0.2116</td>
</tr>
<tr>
<td>8</td>
<td>0.7333</td>
<td>6.5978</td>
<td>9.5067</td>
<td>-0.2211</td>
</tr>
<tr>
<td>9</td>
<td>0.726</td>
<td>6.4874</td>
<td>9.3202</td>
<td>-0.2235</td>
</tr>
<tr>
<td>10</td>
<td>0.74</td>
<td>6.7018</td>
<td>9.5517</td>
<td>-0.2239</td>
</tr>
<tr>
<td>11</td>
<td>0.7828</td>
<td>7.6613</td>
<td>10.0963</td>
<td>-0.2257</td>
</tr>
<tr>
<td>12</td>
<td>0.7704</td>
<td>7.4101</td>
<td>9.9392</td>
<td>-0.2247</td>
</tr>
<tr>
<td>13</td>
<td>0.7941</td>
<td>7.7783</td>
<td>10.2512</td>
<td>-0.2285</td>
</tr>
<tr>
<td>14</td>
<td>0.7821</td>
<td>7.6966</td>
<td>10.0939</td>
<td>-0.2244</td>
</tr>
<tr>
<td>15</td>
<td>0.7978</td>
<td>7.8914</td>
<td>10.2786</td>
<td>-0.2288</td>
</tr>
<tr>
<td>16</td>
<td>0.7823</td>
<td>7.5743</td>
<td>10.0315</td>
<td>-0.2296</td>
</tr>
<tr>
<td>17</td>
<td>0.7885</td>
<td>7.6881</td>
<td>10.1007</td>
<td>-0.2307</td>
</tr>
<tr>
<td>18</td>
<td>0.7934</td>
<td>7.7964</td>
<td>10.1952</td>
<td>-0.2296</td>
</tr>
<tr>
<td>19</td>
<td>0.787</td>
<td>7.6931</td>
<td>10.0844</td>
<td>-0.2298</td>
</tr>
<tr>
<td>20</td>
<td>0.7326</td>
<td>6.6734</td>
<td>9.3758</td>
<td>-0.2241</td>
</tr>
</tbody>
</table>

Table 4.2  Number of Vertices, End Contour Parameters
number of vertices is increased (ranging from 0.6273 to 0.8051 for fewer than 7 vertices and only from 0.726 to 0.7978 for 7 or more vertices.)

Figure 4.2  Probability versus Number of Vertices

The next plot (Figure 4.3) compares the end contour's objective function value versus the number of vertices. Keeping in mind that any \( \nu \)-sided polygon can be thought of as an \((\nu + 1)\)-sided polygon with three of the vertices collinear or
two adjacent vertices identical, we can argue that increasing the number of vertices should monotonically improve (in our case, decrease) the end value of the objective function. Figure 4.3 illustrates that general trend, although there are some minor deviations from that, with the objective function quickly decreasing at first and then remaining fairly constant up to 20 vertices.

Figure 4.3  Objective Function versus Number of Vertices
4.3.2 Finding a Contour of Desired Probability Content

With the results of the preceding section to assure us that polyopt is acting "reasonably" and using them to guide us, we proceed to try to answer some practical questions: how do we find a contour of a desired probability content? While a penalty on the "roughness" of the contour is theoretically necessary, is this necessity exhibited on contours of a small number of vertices? Does this depend on the number of vertices? We will look at the last two of these questions first, examining some overlaid plots of given probability content.

It is natural to think of a smooth figure when considering density contours of a continuous distribution—since in our approach, only polygons are candidate contours, our instinct would have us increase the number of vertices until the resulting contour looks "smooth enough." The problem with this, besides the fact that each additional vertex adds two dimensions to the optimization problem, is that even a contour of a modest number of vertices, e.g., 16, can get pretty messy, as evidenced in Figure 4.4, below.

The contours overlaid on this plot are the contours with 25, 50, and 75 percent empirical probability content (the solid, dotted, and dashed lines, respectively,) where each has 16 vertices and there was no penalty on perimeter (i.e., $\lambda_2 = 0.0$.) This dramatically illustrates the need to include some mechanism for steering the optimization toward smoother contours. While the contours in Figure 4.4 all have the target probability content and do so with minimum area among polygons with
Figure 4.4  Penalty on Perimeter= 0.0, Number of Vertices= 16—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
the same number of vertices and perimeter, it is unlikely that anyone would have them in mind when imagining density contours.

At the other end of the scale (with respect to number of vertices,) consider when the polygon has seven vertices, as in Figure 4.5. We observe that as long as our desired probability content is big enough, in this case 75 percent, we don’t really see the need to penalize roughness. However, in the 50 percent contour and especially in the 25 percent contour, the estimates found without regard for perimeter are not what our instincts would tell us they should be.

Hidden in this second plot is a reason, even with a small number of vertices, to include a penalty on perimeter. When doing the runs of polyopt to find these contours, we discovered that the routine would occasionally turn the candidate contour “inside out.” What would happen is that the penalty on area would push all but one or two of the vertices to nearly the same location, and then on a subsequent iteration, polyopt would move the vertices in such a fashion that a valid polygon resulted, but with vertices in numerical order running clockwise. The result of this was (because of the way area is calculated) polyopt believed it had found a contour with negative area, and because the probability content calculation doesn’t depend on the orientation of the vertices, it would proceed to make the contour larger ad infinitum. This is easily avoided by simply taking the absolute value of the area of the polygon before adding it into the objective function.
Figure 4.5  Penalty on Perimeter= 0.0, Number of Vertices= 7—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Lest the plots above give the impression that *polyopt* cannot find a decent contour, the next two plots give the end contour estimates for $\lambda_2 = 0.04$ for 7 vertices (Figure 4.6) and 16 vertices (Figure 4.7,) respectively. It should be noted that in spite of being jagged, the contours found in the case of 7 vertices look reasonable, and in the case of 16 vertices look very good.

**Figure 4.6** Penalty on Perimeter= 0.04, Number of Vertices= 7—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure 4.7  Penalty on Perimeter= 0.04, Number of Vertices= 16—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
The plots showing all combinations of $\lambda_2$'s, numbers of vertices, and probability contents can be found in section C.2.

These plots were arrived at by fixing the penalty on perimeter and the number of vertices and then finding the penalty on area which gave the appropriate probability content (as we would expect in a Lagrange multiplier situation.) For example, for 10 vertices, the data in Table 4.1 were used to give starting guesses for each of $\lambda_2 = 0.0, 0.02$, and 0.04. Because of the monotone decrease in probability as the penalty on area is increased, finding $\lambda_1$ was a matter of repeated interval halving. Some example results of this process follow in the next two plots—each of the lines plotted is for a fixed number of vertices and penalty on perimeter. For the three probability values 0.25, 0.50 and 0.75 (on the $x$–axis,) the three $\lambda_1$'s (on the $y$–axis) were determined as described above.

The first plot (Figure 4.8) shows the values of $\lambda_1$ when the number of vertices is 13. The solid line is for $\lambda_2 = 0.0$, with the dotted and dashed for 0.02 and 0.04, respectively.

Two things stand out starkly when considering this plot—first, regardless of the penalty on perimeter, estimation of a contour of large (> 50%) empirical probability content is easier to do than that of a small probability content. This can be seen by the greater negative slope between 0.50 and 0.75 then between 0.25 and 0.50. The second point to be considered is that for small probability contours, we would prefer to not have any penalty on perimeter at all. However, we have
Figure 4.8  Number of Vertices= 13—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
already noted that while the calculated probability content may be exactly correct, the result polygon looks nothing at all like a probability contour.

The second plot, Figure 4.9, shows the $\lambda_1$'s determined for the various numbers of vertices (= 7, 10, 13, and 16 as the solid, dotted, short-dashed, and long-dashed lines, respectively,) , but all with fixed penalty on perimeter $\lambda_2 = 0.04$ This example was chosen because it is representative of all of this type: when regarding only probability content and not the shape of the resulting contour, there is no number of vertices which stands out as an obvious choice.

If any useful information could be gathered from this plot, it might be that for this particular data set, the number of vertices ought to be 13 for goal contours of small probability content. This comment also brings up the point that these contours reflect strongly the empirical distribution of the data: different samples could give different results.

With that caveat in mind, we proceed to give some preliminary suggestions regarding choice of number of vertices and penalty on perimeter. For small probability content, it appears from the example that a small number of vertices with a large penalty on perimeter. This reflects intuition as follows: for a small proportion of the points, a smaller contour (and perhaps fewer vertices) may be used; this is true especially if, as would be the case in any practical application of this, the data were sphered and centered before any contour estimation were tried. The larger the penalty on perimeter, the more regular (in the mathematical sense) the result-
Figure 4.9  Penalty on Perimeter = 0.04—Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
ing polygon, and except in the most unusual of cases, the true contours of small probability content are approximately circular. For contours of large probability content, the opposite is true: more vertices allow greater flexibility for showing structure in the data, and smaller penalties on perimeter allow the optimization leeway to explore those possibilities.
4.3.3 Effect of Different Starting Guesses

If polyopt worked ideally, i.e., always found the absolute minimum value of the objective function, and the probability and area components of the objective function, as expressed in FORTRAN, returned exactly the values we need them to (without need of auxiliary routines to avoid pathological cases,) there would be no point in varying our initial contour estimate and examining the end result, because it would always be the same. However, as illustrated in above sections, the way these routines work does leave room for end contour estimates that differ slightly, and we wanted to determine how they behave if a smaller (or larger) radius is used as a starting estimate, or if the initial guess is “off-center,” i.e., has center different from the coordinates of the theoretic mean of the data. Both of these were considered, along with the effect of different \((n = 100, 200, 500, \text{ and } 1000)\) sample sizes. (It should be pointed out that these data sets are supersets, respectively, of one another; i.e., the first 100 ordered pairs of each of these sets are the same, the first 200 of the last three are the same, etc.)

These issues are of import—perhaps not in setting where our sole interest is estimation of a contour of 90% empirical probability content, where any reasonable procedure would first center (move observed mean to \((0, 0)\)) and normalize, or sphere, the data (multiply the two-dimensional data by the inverse of its observed variance-covariance matrix) so that questions of location and scale would be moot. In this instance they are of interest in allowing us to investigate two possible root
causes of different estimates resulting from different initial guesses: existence of multiple local minima and accuracy of the \textit{pdsopt} portion of \textit{polyopt} in finding local minima. We interpret the results (both the end contour and its objective function value) of \textit{polyopt} in this fashion:

1. Much different \iff multiple local minima

2. Slightly different \iff either:

   (a) many local minima (objective function oscillates between them)

   (b) convergence tolerance set too high

These seeming impediments to the functionality of the algorithm can be useful in practice ("it’s not a bug, it’s an undocumented feature.") In situation 1 above, either clustering of the data or large "flat" regions of empirical probability density could be suggested by the algorithm’s inclusion of a subset of the sample for some starting guess, and exclusion of that same subset for another starting guess. In situation 2, end results may indicate two possible solutions: increase the "smoothing" parameter on $h_b$, the bandwidth of the boundary kernel; or decrease the convergence tolerance, the absolute difference between different values of the objective function that \textit{pdsopt} considers inconsequential.
Effect of radius of initial polygon

In all the runs whose results we have reviewed to this point, the starting radius of the estimating polygon has been 1.3, a value chosen almost at random. In this portion of our work, we consider starting radii of 0.6 and 3.0 in comparison to our "standard" value of 1.3 (all with \( \nu = 10, \lambda_1 = 0.02 \) and \( \lambda_2 = 0.04 \).) For each of these three values of the starting radius, *polyopt* was run to completion for the four sample sizes mentioned above. The ubiquitous table giving the end contour parameters follows (Table 4.3.) Encouraging is that within a given sample size, the probabilities and objective function values are close, with the notable exception of the \( n = 200 \) case.

For a given sample size, all three end contours were plotted, together with the data on which they were based. Within each plot, the end contour whose initial

<table>
<thead>
<tr>
<th>( n )</th>
<th>Initial Radius</th>
<th>Probability</th>
<th>Area</th>
<th>Perimeter</th>
<th>Obj. Fcn</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.6</td>
<td>0.8485</td>
<td>8.0028</td>
<td>10.5832</td>
<td>-0.2651</td>
</tr>
<tr>
<td>100</td>
<td>1.3</td>
<td>0.8505</td>
<td>7.9462</td>
<td>10.3872</td>
<td>-0.2761</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>0.8309</td>
<td>7.5032</td>
<td>10.2843</td>
<td>-0.2694</td>
</tr>
<tr>
<td>200</td>
<td>0.6</td>
<td>0.7168</td>
<td>6.367</td>
<td>9.2466</td>
<td>-0.2196</td>
</tr>
<tr>
<td>200</td>
<td>1.3</td>
<td>0.74</td>
<td>6.7018</td>
<td>9.5517</td>
<td>-0.2239</td>
</tr>
<tr>
<td>200</td>
<td>3</td>
<td>0.808</td>
<td>8.1631</td>
<td>10.6475</td>
<td>-0.2189</td>
</tr>
<tr>
<td>500</td>
<td>0.6</td>
<td>0.7912</td>
<td>8.2196</td>
<td>10.4669</td>
<td>-0.2081</td>
</tr>
<tr>
<td>500</td>
<td>1.3</td>
<td>0.7683</td>
<td>7.7588</td>
<td>10.1169</td>
<td>-0.2084</td>
</tr>
<tr>
<td>500</td>
<td>3</td>
<td>0.7538</td>
<td>7.5251</td>
<td>9.9612</td>
<td>-0.2048</td>
</tr>
<tr>
<td>1000</td>
<td>0.6</td>
<td>0.7683</td>
<td>8.2213</td>
<td>10.4828</td>
<td>-0.1846</td>
</tr>
<tr>
<td>1000</td>
<td>1.3</td>
<td>0.7464</td>
<td>7.7238</td>
<td>10.1128</td>
<td>-0.1874</td>
</tr>
<tr>
<td>1000</td>
<td>3</td>
<td>0.7244</td>
<td>7.3048</td>
<td>9.8301</td>
<td>-0.1851</td>
</tr>
</tbody>
</table>

Table 4.3  Sample Size, Initial Radius, and End Contour Parameters
guess had radius 3.0 is given with a solid line, that whose initial guess had radius 1.3 is given with a dotted (···) line, and the dashed line is that resulting from an initial estimate of radius 0.6.

Of note in these plots is only in the \( n = 200 \) case does the contour whose initial estimate had radius 3.0 end up having the largest area—in fact, for the other three cases, the area of the end contour is monotone decreasing with the radius of the initial contour. For the \( n = 100 \) case (Figure 4.10,) we note that the largest initial guess ended as the smallest end contour, and that the other two initial guesses are nearly indistinguishable with respect to area and probability content.

In the plot of the \( n = 200 \) case, we see an artifact of the data which most likely led \textit{polyopt} to leave the contour whose initial guess had radius 3.0 with a larger end radius, larger probability content, and thereby a nearly identical value for the objective function. In the lower right of Figure 4.11, we see that there are approximately 11 points that are included in that contour that are omitted from the other two. We conclude in this case that of the possible settings for this sort of result given above, 1 is operative with strong certainty.

In the \( n = 500 \) case, the three contours are in as close agreement as for any sample size. They agree very closely in overall shape, with the end contours reversing their initial guesses—i.e., the smallest ended up largest, and the largest ended up smallest.
Figure 4.10 End Contours with Different Starting Radii, n=100—Solid Line, r = 3.0; Dotted, r = 1.3; Dashed, r = 0.6
Figure 4.11  End Contours with Different Starting Radii,
\( n = 200 \)—Solid Line, \( r = 3.0 \); Dotted, \( r = 1.3 \); Dashed, \( r = 0.6 \)
Figure 4.12  End Contours with Different Starting Radii, 
$n = 500$—Solid Line, $r = 3.0$; Dotted, $r = 1.3$; Dashed, $r = 0.6$
Good agreement is found between the contours in the final, \( n = 1000 \) case, especially if only those vertices whose \( y \)-coordinate is non-positive are considered. Unfortunately, the entire contour must be considered, and we see the same phenomenon observed in the \( n = 100 \) and \( n = 500 \) cases: good agreement in overall shape, however, the largest contour ended up the smallest, and the smallest ended up the largest.

**Effect of center of initial polygon**

For the runs in this section, we return to a fixed starting radius of 1.3 so that we can observe how changing the center of the initial contour estimate will alter the end contour. To do this, we did an initial run with center at \((0,0)\), and then moved this to \((0.5,0)\) and finally \((1.0,0)\). No other direction was considered because of the symmetry of the theoretic distribution of the data, and no displacement greater than 1.0 is presented because beyond that, polyopt arrived at degenerate contours (those of zero area.) This was repeated, as in the previous section, for sample sizes \( n = 100, \ldots, 1000 \). The outcome we were guarding against was an end contour's location reflecting, to the detriment of its shape, probability content, or objective function value, its initial location.

The table below (4.4) indicates, as in the case of different starting radii in the previous section, that with the exception of the \( n = 200 \) case, the end contour parameters do not indicate much difference within the given sample sizes. Probability
Figure 4.13  End Contours with Different Starting Radii, 
n = 1000—Solid Line, \( r = 3.0 \); Dotted, \( r = 1.3 \); Dashed, \( r = 0.6 \)
and objective function figures are fairly close in these other cases. Unlike the previous section, however, in the \( n = 200 \) case, the increased area in a single instance is not balanced out by a correspondingly increased probability content.

We take the opportunity with this first plot (Figure 4.14) to point out that the solid line is the end contour whose initial guess was centered at \((1.0, 0)\), the dotted line \((0.5, 0)\), and the dashed line \((0, 0)\). The only comment made specifically about this figure is to point out that "middle" of these three, whose end contour is drawn with a dotted line, ended up (examining all three on their left ends, where they differ the most) the farthest to the right, which supports an argument against the "slight" change of location affecting the end contour.

<table>
<thead>
<tr>
<th>( n )</th>
<th>Center (( x )-coord.)</th>
<th>Probability</th>
<th>Area</th>
<th>Perimeter</th>
<th>Obj. Fcn</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0</td>
<td>0.8505</td>
<td>7.9462</td>
<td>10.3872</td>
<td>-0.2761</td>
</tr>
<tr>
<td>100</td>
<td>0.5</td>
<td>0.8248</td>
<td>7.4785</td>
<td>10.2253</td>
<td>-0.2662</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>0.8603</td>
<td>8.1889</td>
<td>10.7868</td>
<td>-0.265</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>0.74</td>
<td>6.7018</td>
<td>9.5517</td>
<td>-0.2239</td>
</tr>
<tr>
<td>200</td>
<td>0.5</td>
<td>0.8125</td>
<td>9.0643</td>
<td>11.2214</td>
<td>-0.1823</td>
</tr>
<tr>
<td>200</td>
<td>1</td>
<td>0.7003</td>
<td>6.1776</td>
<td>9.3871</td>
<td>-0.2012</td>
</tr>
<tr>
<td>500</td>
<td>0</td>
<td>0.7683</td>
<td>7.7588</td>
<td>10.1169</td>
<td>-0.2084</td>
</tr>
<tr>
<td>500</td>
<td>0.5</td>
<td>0.7846</td>
<td>8.0921</td>
<td>10.4585</td>
<td>-0.2044</td>
</tr>
<tr>
<td>500</td>
<td>1</td>
<td>0.7743</td>
<td>7.9377</td>
<td>10.391</td>
<td>-0.1999</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
<td>0.7464</td>
<td>7.7238</td>
<td>10.1128</td>
<td>-0.1874</td>
</tr>
<tr>
<td>1000</td>
<td>0.5</td>
<td>0.7473</td>
<td>7.8818</td>
<td>10.2253</td>
<td>-0.1807</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
<td>0.7347</td>
<td>7.5527</td>
<td>10.1534</td>
<td>-0.1776</td>
</tr>
</tbody>
</table>

Table 4.4  Samples Size, Initial Center, and End Contour Parameters
Figure 4.14  End Contours with Different Starting Centers, \( n = 100 \)—Solid Line, \((1.0, 0)\); Dotted Line, \((0.5, 0)\); Dashed Line, \((0, 0)\)
In what is proving to be the troublesome case of the four different sample sizes, with \( n = 200 \), the case (Figure 4.15) where the starting center is 0.5 (dotted line) concludes with the additional points in the lower right included in the contour. While this is not disconcerting if viewed from the perspective that what we are concerned with finding in these runs whether different end results may be due to poor initial guess, it defies simple explanation in terms of how the objective function is defined and how \textit{polyopt} arrives at a solution.

As was true in the previous section, the \( n = 500 \) case depicted in Figure 4.16 gave the best overall agreement without regard to beginning center. Examination of the probability figures in Table 4.4 above shows all three to be within 0.0163 of one another, or about 8 “whole points” out of 500. Tacit in remarking on this agreement is that there is no evidence of the initial center’s location in the positive \( x \)-axis direction affecting the end result.

Similarly to the \( n = 1000 \) case when examining different starting radii, vertices where the \( y \)-coordinate is less than or equal to zero show agreement regardless of starting center to a greater extent than do those above the \( x \)-axis. In this case, however, there is some evidence that with the greater sample size, the initial displacement of the estimate may affect the end contour. Examination of Figure 4.17, specifically in the upper left of the three contour estimates, reveals them to be in the same order, left to right, as their initial guesses.
Figure 4.15  End Contours with Different Starting Centers,
$n = 200$—Solid Line, $(1.0, 0)$; Dotted Line, $(0.5, 0)$; Dashed Line, $(0, 0)$
Figure 4.16  End Contours with Different Starting Centers, $n = 500$—Solid Line, (1.0, 0); Dotted Line, (0.5, 0); Dashed Line, (0, 0)
Figure 4.17  End Contours with Different Starting Centers, 
n = 1000—Solid Line, (1.0, 0); Dotted Line, (0.5, 0); Dashed Line, (0, 0)
4.3.4 Comparison to Kernel-Based Density Contour Estimates

The simulation work with the optimization-based density contour estimates in the previous three sections has been primarily self-referential in nature, i.e., we have merely run the optimization to conclusion with different starting parameters and, lacking "truth" to compare them to, resorted to comparison of the runs among themselves. There is a component of that nature in what follows in this section, as well, in that part of what we attempt to grasp with the analyses herein is an "average" behavior of the algorithm. By this, we mean to ascertain what happens if the same starting contour, the same penalties on area and perimeter, and the same underlying theoretic distribution are used repeatedly, with the changes from run to run being the actual observations upon which *polyopt* will base its optimization of the density contour estimate (25 different trials for each sample size, \( n \)) and three sample sizes, \( n = 100, 200, \) and 500. This remains self-referential, in that there is no analytic result to point to judge the results here as "good" or "bad."

In order to try to introduce some objectivity, we will compare the end contours derived by *polyopt* and kernel-based density contour estimates to truth. These contours were computed in the following fashion: once a optimization-based contour was calculated, its probability content was found and called \( \phi \), say. A kernel density estimate was then calculated at each of the, for example, \( n = 200 \) replicates of a standard bivariate normal making up the data sets in this section. These were ordered, from least to greatest, say \( \delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(200)} \). We approximate the level
(for the kernel-based density contour estimate) by $\delta_{200(1-\phi)}$ (the $200\phi^{th}$ largest order statistic of the kernel density estimates at the observations.) This was done to use the empirical probability content, rather than kernel density probability content, which would be smaller due to the portions of the kernels falling outside of the contour. With this target value calculated, an evenly-spaced, $26 \times 26$ grid of kernel density estimates over the $x$ and $y$ ranges of the data was calculated and the S-Plus routine contour was used to calculate the appropriate contour. With the optimization- and kernel-based contours determined, "truth" was established as the circular contour (because of the symmetry of the underlying distribution) with radius, $r$, where $r = \sqrt{\chi^2_{2,\phi}}$.

With the process described for finding a density level (and its contour) corresponding to an output estimate from polyopt, we have an exterior reference for judging our procedure. The metric decided on for comparison of the two was the area of the symmetric difference of each of the two contours with "truth," expressed as a proportion of the true contour's area. While the remainder of the discussion will be focused on comparison of these symmetric differences (and a small aside looking at the average probability content as the number of data points is increased,) the plots of the optimization- and KDE-based contours for the $n = 200$ have been included as Figures C.52-C.76 in Appendix C.4. Of interest is that the kernel-based estimate's contour area is, without exception, larger than that of the contour from the optimization-based procedure. A plausible explanation for this is, while the
kernel-based contour estimate level was arrived at by matching probabilities, the contour itself is calculated based on all data values within its bandwidth (with a Gaussian kernel, this would theoretically be all the observations, however, points beyond a certain distance from the contour could be (rightly) argued to have no influence on the kernel-based estimate of the density.) Contrast this to the contour derived by polyopt—optimum in the sense of maximizing probability for the given penalties on area and perimeter—therefore, we should not be surprised when we find that the contours derived from kernel-based estimates are larger in area than those from the optimization-based procedure. That having been said, it should be mentioned that the agreement is, in no instance, bad.

In order to judge the accuracy of our contour estimates computed both using kernel density estimates and from optimized polygons, we ran three experiments of 25 trials each. All 75 trials consisted of i.i.d. replications of bivariate normal data with mean \((0,0)\) and the identity variance-covariance matrix. The first 25 had \(n = 100\) replications, the second, \(n = 200\); and the third, \(n = 500\). Before looking at those accuracy results, however, we give the brief aside promised earlier regarding average probability content as the sample size is increased. The optimization behaved as expected—as sample size was increased, the less variable was the probability content of the end contour. These results are given pictorially below, in the histograms of the probability contents for the 25 different random data
sets for each of the three samples sizes (Figures 4.18 through 4.20, noting that all three have the same horizontal scale.)

**Figure 4.18** Probability Contents for $n = 100$
Figure 4.19  Probability Contents for $n = 200$
Figure 4.20  Probability Contents for $n = 500$
After another demonstration of the reasonability of the optimization’s results, we finally get to evidence of their accuracy. Figures 4.21, 4.22, and 4.23 are bivariate plots of the outcomes of the \( n = 100, 200, \) and \( 500 \) runs, respectively. Each plot has symmetric differences expressed as a proportion of the area of the true contour as its axes, with difference from optimization-based on the \( y \)-axis and difference from the kernel-based estimate on the \( x \)-axis. Also on each plot is included the identity line, \( y = x \), for reference. The result we strive for as a proponent of the optimization-based method would be for all plotted points to lie below that line, indicating that it had come closer to matching truth than did those contours arising from the kernel-based approach.
Figure 4.21 Symmetric Difference of Optimization and KDE-Based Contours with Truth
As you can see, our goal was not realized—in fact, the results seen in Figure 4.21 are typical of the other two sample sizes. For $n = 100$, out of the 25 cases, in only 9 of them was the area of the symmetric difference with the optimization-based contour smaller than with the kernel-based contour. For the other two sample sizes, 7 and 8 were smaller.

**Figure 4.22** Symmetric Difference of Optimization and KDE-Based Contours with Truth
Figure 4.23 Symmetric Difference of Optimization and KDE-Based Contours with Truth
Happily, there are a couple of bright spots to be found, even in these results: the first of these is that even as the proportion of the time the optimization-based contours come closer to truth than their kernel based counterparts remains fairly constant, we do see the distribution of these differences getting smaller as the sample size increases. This result goes well with the result cited above regarding the average probability content varying less as the sample size increased. The second bright spot is summarized in a tabular form below: while a "sign test" analysis of optimization- versus kernel-based contour estimates as was done above would not bode well for the former, when the magnitudes of the differences are considered as well, the two look quite comparable. First, if we consider the mean proportion of true area the symmetric differences represent, then Table 4.5 casts a brighter light on the optimization method, with the mean proportion differing by .03 in the $n = 100$ case, and by .019 and .006 in the cases of the larger sample sizes.

<table>
<thead>
<tr>
<th>$n$</th>
<th>KDE-Based</th>
<th>Optimization-Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.2073319</td>
<td>0.2374085</td>
</tr>
<tr>
<td>200</td>
<td>0.1526031</td>
<td>0.1717737</td>
</tr>
<tr>
<td>500</td>
<td>0.1101186</td>
<td>0.1163736</td>
</tr>
</tbody>
</table>

**Table 4.5** KDE- and Optimization-Based Contours—Mean Symmetric Differences with Truth
When medians are examined (Table 4.6), there is nearly a constant difference of .011 between the two methods—smaller (than the mean difference) in the two smaller sample sizes, but nearly twice the mean difference in the case of the larger sample size.

<table>
<thead>
<tr>
<th>n</th>
<th>KDE-Based</th>
<th>Optimization-Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.2069489</td>
<td>0.2188521</td>
</tr>
<tr>
<td>200</td>
<td>0.1465923</td>
<td>0.1561003</td>
</tr>
<tr>
<td>500</td>
<td>0.1047754</td>
<td>0.1159662</td>
</tr>
</tbody>
</table>

Table 4.6 KDE- and Optimization-Based Contours—Median Symmetric Differences with Truth

We conclude that in spite of their different geneses, the optimization- and kernel-based methods reach similar results. Their characteristics mentioned above (especially the fact that those derived from polygon optimization necessarily will be smaller) give us a reasonable basis to accept these outcomes as logical.
4.3.5 Behavior with Asymmetric/Bimodal Data

Asymmetric Data

We first investigate a distribution which is a premiere among the two-dimensional data sets, in that it is not standard bivariate normal: it is a 50/50 (actually 250/250, as far as number of sample points) mixture of two bivariate normal distributions. The first of these had mean \( \begin{pmatrix} 3/4 \\ 0 \end{pmatrix} \) with variance-covariance matrix 
\[
\begin{bmatrix}
2/3 & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\cos(-\pi/4) & -\sin(-\pi/4) \\
\sin(-\pi/4) & \cos(-\pi/4)
\end{bmatrix},
\]
written in this fashion to emphasize that it is merely a standard bivariate normal, squeezed by a factor of 2/3 in the \( x \)-direction, rotated clockwise by \( \pi/4 \), and then shifted 0.75 to the right. The second half of this mixture distribution is the mirror to this one: still squeezed by a factor of 2/3 in the \( x \)-direction, but then rotated counter-clockwise by \( \pi/4 \), and then shifted 0.75 to the left. A sample of size \( n = 500 \), the data used in all of the examples in this section, is pictured in Figure 4.24, below.

The goal was a data set which had a "boomerang" or "banana" structure underlying it without that shape being so obvious that the optimization routine couldn't help but stumble upon it. The reader is left to decide whether the data as pictured in Figure 4.24 meets these criteria.

Once these data were generated, polyopt was put to work, and with encouraging results. We take this opportunity to point out that all the contours in this section have 15 vertices. The first two plots (Figures 4.25 and 4.26) show the resulting
Figure 4.24  Asymmetric Data
contour estimate when the penalty on area is set at $\lambda_1 = 0.06$, with the first plot depicting the end contour when $\lambda_2 = 0.012$ and the second when $\lambda_2 = 0.005$. We point out that these values are much smaller than those used above when trying to arrive at contours in the case of symmetric data. The smaller values of the penalty on perimeter allows the end contours to follow more closely patterns in the data. In Figure 4.25, these shows minimally in the small indentation on the top of the contour. Further reduction of the perimeter penalty in Figure 4.26, while more “banana-like,” is also more irregular and less appealing as a contour (especially the indentations at bottom right and top left.)
Figure 4.25  Asymmetric Data
Figure 4.26  Asymmetric Data
The final three contours (plotted in Figures 4.27 through 4.29) have $\lambda_1 = 0.08$ in common, while the penalty on perimeter is set at $\lambda_2 = 0.02$, 0.01, and 0.005, respectively. As in the cases above, the lower our penalty on perimeter, the more the resulting contour estimate reflects the structure in the data.

Figure 4.27  Asymmetric Data
Figure 4.28  Asymmetric Data
Figure 4.29  Asymmetric Data
The conclusion is fairly obvious—we cannot expect our optimization routine to uncover structure due to the underlying distribution in the data (through tweaking of the area and perimeter penalties) without it, at the same time, reflecting structure due to random anomalies in the data set. This fact carries on into the next section, wherein we explore a data set with bimodal tendencies.

Bimodal Data

As mentioned in the introduction of this work, the original motivation in attempting to derive density contours was to automate the investigation of the modality of a given data set. In one dimension, we may conclude that our data is not unimodal if our density estimate procedure returns 3 or more values when searching for values in the domain of the random observations whose density estimate is a pre-determined value. The lower this pre-determined value, the more evidence is being required to rule out unimodality.

In two dimensions, we can no longer use this rule, since our density contour for a particular level will consist of an uncountable union of \((x, y)\) ordered pairs (unless we find that/those unique point(s) whose density estimate is greater than that of all the other points.) So to decide in favor of multimodality in more than one dimension, we look for the higher-dimensional analogue of “3 or more values”—disconnected regions of density estimates at or above a certain level. As in the one-dimensional case, the lower this value (and, anticipating how it will be solved
in the optimization-based procedure, the greater the probability content of the resulting contour estimate, the stronger the evidence for ruling out unimodality.

The plots that follow show data from mixture distributions, in each case one-half (exactly 250 i.i.d. replications, not half "in probability") from a Gaussian distribution with mean \((\Delta x, 0)\) and one-half with mean \((-\Delta x, 0)\), both with variance-covariance matrix \(s \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\). The \(\Delta x\) (separation) and \(s\) (scaling) are specified in each case, and were varied to demonstrate how the optimization-based procedure behaves over a realistic range of distribution types. For all of these, the initial contour estimate was an oval, with center at \((0, 0)\) and major axis of 3 (in the \(x\)-direction) and minor axis \(3s\), with the \(s\) the same as the scaling factor of the variance-covariance matrix for the given trial.

The first plot, Figure 4.30, has \(\Delta x = 1.25\)—for any smaller value, not much separation was evident, and so end contour estimates reflected this: they appeared ovate. The scaling factor is \(s = \frac{2}{3}\). The settings for \(\lambda_1\) and \(\lambda_2\) were 0.06 and 0.008, and the "boundary kernel bandwidth," the distance beyond which points are considered wholly in or out of the contour, \(h_b\), was set at 0.05, half the value used in all preceding runs. With these parameters, we see the first evidence (in the resulting contour estimate) of something other than a unimodal distribution. Note how much smaller the penalty on perimeter was set for this trial—this allowed \texttt{polyopt} to "push" the vertex toward the center.
Figure 4.30  Bimodal Data
The next four plots, Figures 4.31-4.34, were all done with the same set of data, with $\Delta x = 1.5$ and $s$ still set to $\frac{2}{3}$. For these, we returned to $h_b = 0.1$. In Figure 4.31, where $\lambda_1$ and $\lambda_2$ were 0.03 and 0.004, respectively, we see basically the same result as in Figure 4.30, in spite of the increased displacement between the two "half" distributions and the decreased penalty on area and perimeter.

Figure 4.31  Bimodal Data
Concerned that perhaps 10 vertices were not allowing the procedure enough leeway to reflect the bimodality of the data, we increased the number of vertices to 15, leaving all other input parameters and the data the same, and the output of that run follows in Figure 4.32. Here we definitely see structure in our end contour estimate—perhaps not bimodality, but certainly something other than an ovate contour.

![Figure 4.32 Bimodal Data](image_url)
This example was pursued further by increasing only the penalty on area (to 0.04,) leaving the penalty on perimeter unchanged. The outcome of this is depicted in Figure 4.33, with positive and negative results: while the end contour estimate at this point definitely indicates bimodality, we have also ended up with 2 “concave” vertices in the right half contour (besides that in the upper left which serves to link it to the left half) and one concave vertex in the left—in other words, an indicator of bimodality was gained at the cost of some intuitive notion of “smoothness.” To the credit of the procedure, even these drawbacks are supported by intuition in that they occur in areas of low empirical density.

The final trial with this same data set was done with $\lambda_1 = .05$, a further increase, to see whether further demarcation occurred, or whether, as was the case with many settings on the penalties, the contour settled on one of the modes, ignoring the other entirely. The result, shown in Figure 4.34, does show further separation of the two half contours, and the half contours’ “smoothness” has improved as well.

Our last two figures are illustrative for the cases where the separation is total, and are of interest primarily because they give an example of how the setting of $h_b$, the boundary kernel bandwidth, can effect the end contour using this procedure. For both of these, we return to 10 vertices, $\Delta x = 2$, $s = \frac{1}{3}$, $\lambda_1 = 0.08$ and $\lambda_2 = 0.01$, slightly larger than in the cases immediately preceding, because in spite of the greater relative displacement, the vertical scale has been halved while the
Figure 4.33  Bimodal Data
Figure 4.34  Bimodal Data
horizontal remained the same. For this first plot, Figure 4.35, we set $h_b = 0.05$—a positive outcome with respect to reflecting the obvious bimodality of the data, however the right half contour, with its "empty" corner (lower left) necessitated by remaining connected to the left half contour, doesn't bear much resemblance to a density contour.

![Figure 4.35 Bimodal Data](image-url)
Figure 4.36, our last, is the plot of the end contour for the same data set with all input parameters to polyopt unchanged save $h_b$, now set at 0.1 again. Separation is no less definite, however the two half contours definitely “look” better.

Figure 4.36  Bimodal Data
Chapter 5

Concluding Remarks

We decide in the end that optimization-based contour estimation is feasible, giving results which are internally consistent, comparable to those derived from widely used methods, and that may indicate the presence of structure or more than one mode in the data.

Issues remaining for later study include, but are obviously not limited to:

- Consideration of other measures of “roughness”—perimeter was used in this spot for all studies whose analyses were included here, however it’s unavoidable linkage to area is a weakness. Another possible measure includes a chi-squared statistic on the exterior angles: a regular $n$-gon would give a value of 0, since all the angles are the same and therefore equal the mean

- Theoretic calculation of the best values for the penalties on area and roughness for a given goal: whether that is a specific probability content in the end contour or detection and indication of multimodality

- Exploration of an iterative method, where the boundary kernel bandwidth, $h_b$, is gradually pared down, so the end contour is exactly that polygon of minimum area/roughness combination for the given probability content
• As in the previous point, an iterative method, only where the number of vertices is allowed to increase to a point where the

(prob. content, area, roughness)

triple remains "constant"

• Exploration of the mode-hunting potential of the procedure—this would involve making the improvements suggested in the previous bullets as well as definition of "branch points" in the procedure, i.e., decision rules for when to split contours into separate polygons and how to iteratively call the procedure to separately optimize the resulting "pieces"
Chapter 6

Proofs of Theorems

6.1 Properties of $\hat{f}_n$

Because the $X_i$ are independent and identically distributed, the random variables $K\left(\frac{x-X_1}{h_n}\right)$ are also i.i.d. This is a key fact used in both of the parts below.

1. Expectation of $\hat{f}$

$$
E\hat{f}_n(x) = h_n^{-1}E\left[ K\left(\frac{x-X_1}{h_n}\right) \right]
= h_n^{-1} \int K\left(\frac{t-x}{h_n}\right) f(t) \, dt
= \int K(u)f(x+hnu) \, du
= \int K(u)f(x) \, du + \int K(u)f'(x)(uh_n) \, du + \frac{1}{2} \int K(u)f''(x)(uh_n)^2 \, du + \frac{1}{6} \int K(u)f'''(x)(uh_n)^3 \, du + \cdots
= f(x) + 0 + \frac{1}{2}h_n^2 f''(x)\sigma_K^2 + O(h_n^4)
$$

2. Variance of $\hat{f}$

$$
\text{Var} \hat{f}_n(x) = (nh_n^2)^{-1} \left[ EK^2 \left( \frac{x-X_1}{h_n} \right) - \left[ EK \left( \frac{x-X_1}{h_n} \right) \right]^2 \right]
$$
We use the result of part (1) for the second term. For the first, we again use a Taylor series expansion.

\[ h_n^{-1}EK^2 \left( \frac{x - X_1}{h_n} \right) = h_n^{-1} \int K^2 \left( \frac{t - x}{h_n} \right) f(t) \, dt \]
\[ = \int K^2(u)f(x + uh_n) \, du \]
\[ = \int K^2(u)f(x) \, du + \int K^2(u)f'(x)(uh_n) \, du \]
\[ + \frac{1}{2} \int K^2(u)f''(x)(uh_n)^2 \, du + \cdots \]
\[ = f(x)R(K) + 0 + O(h_n^2) \]

And so:

\[ \text{Var} \hat{f}_n(x) = (nh_n)^{-1} \left[ f(x)R(K) + O(h_n^2) - h_n^{-1} \left[ h_n f(x) + O(h_n^3) \right]^2 \right] \]
\[ = (nh_n)^{-1} [f(x)R(K) - h_n f^2(x) + O(h_n^2)] \]
\[ = (nh_n)^{-1} f(x)R(K) - n^{-1} f^2(x) + O(h_n n^{-1}) \]

### 6.2 Royden exercise

\( \Rightarrow \) \( x_n \to x \) implies given \( \epsilon > 0 \) \( \exists N \ni \lvert x_n - x \rvert < \epsilon \) when \( n \geq N \). This implies if \( \langle x_{n_j} \rangle \) is any subsequence of \( \langle x_n \rangle \), then \( \lvert x_{n_j} - x \rvert < \epsilon \) when \( n_j > N \), meaning the subsequence converges to \( x \). This means every subsequence of \( \langle x_n \rangle \) converges to \( x \) and similarly, every sub-subsequence converges to \( x \). Therefore \( x_n \to x \) implies every subsequence of \( \langle x_n \rangle \) has in turn a subsequence which converges to \( x \).
(\Leftrightarrow) Suppose \( x_n \not\to x \). This would imply that \( \exists \ \epsilon > 0 \) for which there are infinitely many terms of \( \langle x_n \rangle \) outside of \( [x-\epsilon, x+\epsilon] \); call these terms the subsequence \( \langle x^*_n \rangle \). By assumption, this subsequence has in turn a subsequence which converges to \( x \). This forces a contradiction, since the entire subsequence \( \langle x^*_n \rangle \) is outside of \( [x-\epsilon, x+\epsilon] \). Therefore, \( x_n \to x \).

### 6.3 One-Dimensional Density Contour Estimation

By assumption \( f'(x_\alpha) \neq 0 \); this along with the existence and absolute continuity of \( f'' \) (in \( \mathcal{F} \)) implies \( \exists \ \gamma > 0 \) and \( \epsilon > 0 \) \( \forall \) either \( f'(x) > \gamma \) or \( f'(x) < -\gamma \ \forall \ x \in [x_\alpha - \epsilon, x_\alpha + \epsilon] \). WLOG, let us assume \( f'(x) < -\gamma \) and call \( G = [x_\alpha - \epsilon, x_\alpha + \epsilon] \).

We observe this implies \( f(x_\alpha - \epsilon) \geq \alpha + \gamma \epsilon \) and \( f(x_\alpha + \epsilon) \leq \alpha - \gamma \epsilon \). If we call

\[
A^c_1 = \{ \hat{f}_n(x_\alpha - \epsilon) < \alpha + \gamma \epsilon / 2 \text{ or } \hat{f}_n(x_\alpha + \epsilon) > \alpha - \gamma \epsilon / 2 \text{ for infinitely many } n \},
\]

then \( A^c_1 \subset \{ \sup_{x} |\hat{f}_n - f| \geq \gamma \epsilon / 2 \text{ for infinitely many } n \} \) and by Theorem 3.2, since \( \sup_{x} |\hat{f}_n - f| \xrightarrow{\text{a.s.}} 0 \), the probability of this last event is necessarily 0. Therefore,

\[
1 = P(A_1) = P\{ \exists N \ \exists n > N, \ \hat{f}_n(x_\alpha - \epsilon) \geq \alpha + \gamma \epsilon / 2 \\
\quad \text{and } \hat{f}_n(x_\alpha + \epsilon) \leq \alpha - \gamma \epsilon / 2 \} \tag{6.1}
\]

\[
\leq P\{ \exists N \ \forall n > N, \ \exists \tilde{x}_{na} \in G \ \exists \hat{f}_n(\tilde{x}_{na}) = \alpha \}
\]

If we call

\[
A^c_2 = \{ \hat{f}_n(x) \geq 0 \text{ for some } x \in G \text{ for infinitely many } n \}
\]
\[
\{ \sup_x |\hat{f}^n_n - f| > \gamma \text{ for infinitely many } n \} \\
\]

This time by Theorem 3.3, since \( \sup_x |\hat{f}^n - f'| \overset{a.s.}{\rightarrow} 0 \), the probability of this last set is 0 and we get

\[
1 = P(A_2) = P\{ \exists N \ \forall n > N \ \hat{f}(x) < 0 \ \forall x \in G \}. 
\tag{6.2}
\]

If we then call \( A = A_1 \cap A_2 \), then by 6.1 and 6.2, we get

\[
1 = P(A) = P\{ \exists N \ \forall n > N \ \exists \hat{x}_{n_\alpha} \in G \ \exists \hat{f}_n(\hat{x}_{n_\alpha}) = \alpha \\
\text{and } \hat{f}^n_n(x) < 0 \ \forall x \in G \} \\
\leq P\{ \exists N \ \forall n > N \ \exists! \hat{x}_{n_\alpha} \in G \ \exists \hat{f}_n(\hat{x}_{n_\alpha}) = \alpha \} 
\]

Taking any \( \omega \in A \) (and because \( P(A) = 1 \), define \( \langle \hat{x}_{n_\alpha} \rangle \) arbitrarily on \( A^c \), since this happens with probability 0), consider the sequence of sample solutions, \( \langle \hat{x}_{n_\alpha} \rangle \).

From this take any subsequence \( \langle \hat{x}^*_{n_\alpha} \rangle \). This is an infinite subset of \( G \), a compact set, so by Theorem 3.4, \( \langle \hat{x}^*_{n_\alpha} \rangle \) has a limit point in \( G \). If we call this limit point \( x_L \), this is equivalent to saying there is a further subsequence \( \langle \hat{x}^*_{n_k\alpha} \rangle \) (of \( \langle \hat{x}^*_{n_\alpha} \rangle \)) which converges to \( x_L \). Let us assume \( x_L \neq x_\alpha \). Then \( \exists \delta > 0 \ \exists |x_L - x_\alpha| > \delta \). From our assumption on \( f'(x) \) for \( x \in G \), \( |f(x_L) - f(x_\alpha)| > \delta \gamma \). But by the continuity of \( f \) and the uniform consistency of \( \hat{f}_n \),

\[
\alpha = \hat{f}_n(\hat{x}_{n_k\alpha}) = \lim_{k \to \infty} \hat{f}_n(\hat{x}_{n_k\alpha}) = f(x_L),
\]
a contradiction. Therefore $x_L = x_\alpha$, and we’ve shown that every subsequence of $(\hat{x}_n)$ has, in turn, a subsequence which converges to $x_\alpha$. By Theorem 3.5, this implies $\hat{x}_n \to x_\alpha \forall \omega \in A$.

6.4 Asymptotic Normality of $\hat{f}_n(x_\alpha)$

First we note, using the Mean Value Theorem, that

$$\hat{f}_n(\hat{x}_n) = \hat{f}_n(x_\alpha) + (\hat{x}_n - x_\alpha)\hat{f}'_n(\hat{x}_n)$$

where $\hat{x}_n$ is between $x_\alpha$ and $\hat{x}_n$. We then note that $\hat{f}_n(\hat{x}_n) = \alpha$. Using these to solve for $\hat{x}_n$ in this last display gives us

$$\hat{x}_n = x_\alpha + \frac{\alpha - \hat{f}_n(x_\alpha)}{\hat{f}_n'(\hat{x}_n)}$$

The only random components on the right-hand side of this expression are $\hat{f}_n(x_\alpha)$ and $\hat{f}_n'(\hat{x}_n)$. The first of these, properly standardized, converges to a standard normal by Theorem 3.7. The second converges in probability to $f'(x_\alpha)$ since $\hat{x}_n \xrightarrow{P} x_\alpha$ and $\sup |\hat{f}'_n - f'| \xrightarrow{a.s.} 0$. These two facts along with Slutsky’s Theorem give the desired result.

6.5 Basis Functions for a Rotationally Invariant Process

Let $\{X(t), t \in [0, 2\pi]\}$ be a rotationally invariant $L_2$ process. As pointed out in the main text, this means $\mathbb{E}X(t) = c \forall t \in [0, 2\pi)$ (the constant $c$ may be taken
WLOG to be 0) and $\text{Cov}[X(s), X(t)] = \text{E}[X(s)X(t)] = \kappa(s \ominus t)$ for some function $\kappa()$ of one variable.

The functions $\{\exp(i\nu t), \nu \in \mathbb{Z}\}$ are orthonormal and span $\mathcal{L}_2$, facts well-established in Fourier transform theory. What remains to be shown is that they satisfy the eigenfunction-eigenvalue equation:

$$(K\varphi_{\nu})(t) = \int_0^{2\pi} \text{Cov}[X(s), X(t)]\varphi_{\nu}(s) \, ds = \lambda_{\nu}\varphi_{\nu}(t)$$

But with $\varphi_{\nu}(t) = \exp(i\nu t)$, we get:

$$\begin{align*}
(K\varphi_{\nu})(t) &= \int_0^{2\pi} \text{Cov}[X(s), X(t)]\exp(i\nu s) \, ds \\
&= \int_0^{2\pi} \kappa(s \ominus t) \exp(i\nu s) \, ds \\
&= \exp(i\nu t) \int_0^{2\pi} \kappa(s \ominus t) \exp[i\nu(s - t)] \, ds \\
&= \exp(i\nu t) \int_0^{2\pi} \kappa(s \ominus t) \exp[i\nu(s \ominus t)] \, ds \\
&= \exp(i\nu t) \int \kappa(u) \exp(i\nu u) \, du \\
&= \exp(i\nu t)\lambda_{\nu}
\end{align*}$$

The display following the fourth "=" above follows from the periodicity of the complex exponential.
# Appendix A

## Symbols Used

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.s.</td>
<td>almost sure convergence</td>
</tr>
<tr>
<td>≈</td>
<td>asymptotically equivalent</td>
</tr>
<tr>
<td>$\mathcal{D}$</td>
<td>convergence in distribution</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>convergence in probability</td>
</tr>
<tr>
<td>$\forall$</td>
<td>for all, for each, for every</td>
</tr>
<tr>
<td>$\mathbb{R}$</td>
<td>real numbers</td>
</tr>
<tr>
<td>$\exists$, s.t.</td>
<td>such that</td>
</tr>
<tr>
<td>$\exists$</td>
<td>there exists</td>
</tr>
<tr>
<td>!</td>
<td>unique</td>
</tr>
<tr>
<td>WLOG</td>
<td>without loss of generality</td>
</tr>
</tbody>
</table>
Appendix B

Karhunen-Loève Expansion

The idea of the Karhunen-Loève expansion is expressing, a lă the Fourier series, an $L_2$ process as a sum of random coefficients multiplying orthonormal functions. The appeal of this expansion is shown in section B.2 below: when the expansion is calculated for a Gaussian process, the random coefficients are jointly Gaussian.

B.1 Karhunen-Loève Theorem

Let $\{X(t), a \leq t \leq b\}$, where $a, b$ are finite, be an $L_2$ process with zero mean and continuous covariance $K$ (i.e., $K(s, t) = \text{Cov}[X(s), X(t)] = E[X(s)X(t)]$). Let $\{e_n, n = 1, 2, \ldots\}$ be an orthonormal basis for the space spanned by the eigenfunctions of the nonzero eigenvalues of the integral operator associated with $K$ on $L[a, b]$,

$$(Ax)(s) = \int_a^b K(s, t)x(t) \, dt, \quad a \leq s \leq b, \quad x \in L_2[a, b],$$

with $e_n$ taken as an eigenfunction corresponding to the eigenvalue $\lambda_n$. Then

$$X(t) = \sum_{k=1}^{\infty} Z_k e_k(t), \quad a \leq t \leq b,$$
where $Z_n = \int_a^b X(t)e_n(t) \, dt$, and the $Z_n$ are orthogonal random variables with zero mean and $E[|Z_n|^2] = \lambda_n$. For $t \in [a, b],

E \left[ \left| X(t) - \sum_{k=1}^n Z_k e_k(t) \right|^2 \right] \to 0 \quad \text{as} \quad n \to \infty,

uniformly in $t$.

### B.2 Karhunen-Loève Expansion of a Gaussian Process

In the Karhunen-Loève expansion above for a Gaussian process, the random variables $Z_k$ form a Gaussian sequence, that is, for each $r$, $Z_1, \ldots, Z_r$ are jointly Gaussian. If the random variables of the process are real, the $Z_k$ are independent.
Appendix C

Figures Illustrating Optimization-Based Contours

C.1 Contours for Range of Penalties on Area and Perimeter

All figures which follow in this section show contours minimizing the objective function for the same set of data, but with different values for the penalties on area and perimeter, $\lambda_1$ and $\lambda_2$, respectively. As mention in section 4.3.1, both of these penalties have a natural minimum value of 0 (zero); to find a maximum (for the given data set,) each was increased in increments of 0.02 until a "pathological" case was reached, i.e., the contour arrived at was no longer a polygon, but rather a line segment or connected line segments.
Figure C.1  End Contour with $\lambda_1 = 0.00$ and $\lambda_2 = 0.00 - 0.06$
Figure C.2  End Contour with $\lambda_1 = 0.00$ and $\lambda_2 = 0.08 - 0.12$
Figure C.3  End Contour with $\lambda_1 = 0.02$ and $\lambda_2 = 0.00 - 0.06$
Figure C.4  End Contour with $\lambda_1 = 0.02$ and $\lambda_2 = 0.08 - 0.10$
Figure C.5  End Contour with $\lambda_1 = 0.04$ and $\lambda_2 = 0.00 - 0.06$
Figure C.6  End Contour with $\lambda_1 = 0.04$ and $\lambda_2 = 0.08$
Figure C.7  End Contour with $\lambda_1 = 0.06$ and $\lambda_2 = 0.00 - 0.06$
Figure C.8  End Contour with $\lambda_1 = 0.08$ and $\lambda_2 = 0.00 - 0.06$
Figure C.9  End Contour with $\lambda_1 = 0.10$ and $\lambda_2 = 0.00 - 0.04$
Figure C.10  End Contour with $\lambda_1 = 0.12$ and $\lambda_2 = 0.00 - 0.02$
Figure C.11  End Contour with $\lambda_1 = 0.14$ and $\lambda_2 = 0.00$
C.2 Contours for Various Numbers of Vertices, Penalties on Perimeter

The data plots which follow were all generated in Monte Carlo bivariate trials with mean \( (0,0) \) and variance-covariance matrix \( \begin{pmatrix} 10 \\ 0 \\ 01 \end{pmatrix} \) and the optimization-based contour was found for the specified penalty, \( \lambda_2 \), on perimeter and number of vertices for the polygon. In each trial, \( n = 200 \) i.i.d. replications were used, with the same starting seed for each.

For each plot, the first contour is plotted as a solid line, the next as a dotted line, the third as short dashes, and when present the fourth contour is plotted with long dashes. There are three types of plots: the first twelve are plots where the number of vertices and the penalty on perimeter are held constant on the plot, and \( \lambda_1 \) is found that gives the three contours with empirical probability content 0.25, 0.50, and 0.75 (as the first, second, and third contours, respectively.) All the values for \( \lambda_1 \) which yielded these probabilities are given in Table 4.1. The next twelve plots have the same number of vertices and the same empirical probability content for all contours included thereon, and the penalty on perimeter is varied (over the values 0.0, 0.02, and 0.04)—these were completed to allow us to examine the effect of \( \lambda_2 \) on comparable contours. The final nine plots hold empirical probability content and penalty on area constant and give the contours for the four different setting for the number of vertices, 7, 10, 13, and 16.
C.2.1 Different Probability Contents

Each of the plots in Figures C.12 through C.23 have the approximate 0.25, 0.50, and 0.75 probability content contours overlaid.

Figure C.12 Penalty on Perimeter= 0.0, Number of Vertices= 7—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.13  Penalty on Perimeter = 0.0, Number of Vertices = 10—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.14  Penalty on Perimeter= 0.0, Number of Vertices= 13—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.15  Penalty on Perimeter= 0.0, Number of Vertices= 16—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.16  Penalty on Perimeter = 0.02, Number of Vertices = 7—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.17  Penalty on Perimeter= 0.02, Number of Vertices= 10—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.18  Penalty on Perimeter = 0.02, Number of Vertices = 13—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.19  Penalty on Perimeter= 0.02, Number of Vertices= 16—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.20  Penalty on Perimeter= 0.04, Number of Vertices= 7—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.21  Penalty on Perimeter= 0.04, Number of Vertices= 10—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.22  Penalty on Perimeter= 0.04, Number of Vertices= 13—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
Figure C.23  Penalty on Perimeter = 0.04, Number of Vertices = 16—Solid Line, 25% Contour; Dotted Line, 50% Contour; Dashed Line, 75% Contour
C.2.2 Different Penalties on Perimeter

The three contours overlaid on Figures C.24 through C.35 are for the three different penalties on perimeter, 0.0, 0.02, and 0.04.

**Figure C.24** Number of Vertices= 7, Empirical Probability Content= 0.25—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.25  Number of Vertices= 7, Empirical Probability
Content= 0.5—Solid Line, λ₂ = 0.0; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.26  Number of Vertices = 7, Empirical Probability

Content = 0.75—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.27  Number of Vertices= 10, Empirical Probability
Content= 0.25—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.28  Number of Vertices= 10, Empirical Probability
Content= 0.5—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.29  Number of Vertices= 10, Empirical Probability
Content= 0.75—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.30  Number of Vertices= 13, Empirical Probability Content= 0.25—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.31  Number of Vertices= 13, Empirical Probability
Content= 0.5—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.32  Number of Vertices= 13, Empirical Probability
Content= 0.75—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.33  Number of Vertices = 16, Empirical Probability
Content = 0.25—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.34 Number of Vertices = 16, Empirical Probability
Content = 0.5—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.35  Number of Vertices= 16, Empirical Probability
Content= 0.75—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
C.2.3 Different Numbers of Vertices

These final nine plots have overlaid contours of 7, 10, 13, and 16 vertices for the indicated probability content and penalty on perimeter.

Figure C.36  Penalty on Perimeter= 0.0, Empirical Probability Content=0.25—Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
Figure C.37  Penalty on Perimeter= 0.0, Empirical Probability Content=0.5—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
Figure C.38  Penalty on Perimeter= 0.0, Empirical Probability
Content=0.75—Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed Line, 13;
and Long-Dashed Line, 16
Figure C.39  Penalty on Perimeter $= 0.02$, Empirical Probability Content $= 0.25$—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
Figure C.40  Penalty on Perimeter= 0.02, Empirical Probability
Content=0.25—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed Line, 13;
and Long-Dashed Line, 16
Figure C.41  Penalty on Perimeter= 0.02, Empirical Probability Content=0.75—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
Figure C.42  Penalty on Perimeter= 0.04, Empirical Probability Content=0.25—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
Figure C.43  Penalty on Perimeter= 0.04, Empirical Probability Content=0.5—Solid Line, \( \nu = 7 \); Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
Figure C.44  Penalty on Perimeter= 0.04, Empirical Probability
Content=0.75—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed Line, 13;
and Long-Dashed Line, 16
C.3 Penalties on Area for Desired Probabilities

The first three plots are for fixed penalties on perimeter, with different lines overlaid for 7, 10, 13, and 16 vertices (the solid, dotted, dashed, and long-dashed lines, respectively.) The target probabilities at which $\lambda_1$ was determined were (in all plots): 0.25, 0.50, and 0.75.
Figure C.45  Penalty on Area when Penalty on Perimeter=0.0—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
Figure C.46  Penalty on Area when Penalty on Perimeter=0.02—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
Figure C.47  Penalty on Area when Penalty on Perimeter=0.04—Solid Line, $\nu = 7$; Dotted Line, 10; Short-Dashed Line, 13; and Long-Dashed Line, 16
The next four plots (Figures C.48 through C.51) have the number of vertices fixed within each plot, with a line for each of the values of $\lambda_2 = 0.0, 0.02, \text{ and } 0.04$ (solid, dotted, and dashed, respectively.)

**Figure C.48** Penalty on Area when Number of Vertices = 7—Solid Line. $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.49  Penalty on Area when Number of Vertices=10—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.50  Penalty on Area when Number of Vertices= 13—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
Figure C.51  Penalty on Area when Number of Vertices= 16—Solid Line, $\lambda_2 = 0.0$; Dotted Line, 0.02; Dashed Line, 0.04
C.4 Contours for Different Random Data Sets

The data plots which follow were all generated in Monte Carlo bivariate trials with mean \( \begin{pmatrix} 0 \\ 0 \end{pmatrix} \) and variance-covariance matrix \( \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \) and the optimization-based contour was found for penalties \( \lambda_1 = 0.02 \) on area and \( \lambda_2 = 0.04 \) on perimeter. In each trial, \( n = 200 \) i.i.d. replications were used, with the starting seed for the random number generator indicated beneath each. The smoother contour in each figure is that based on a kernel density estimate as described in section 4.3.4.
Figure C.52  Seed = 17643
Figure C.53  Seed = 29221
Figure C.54  Seed = 15565
Figure C.55  Seed = 20145
Figure C.57  Seed = 13663
Figure C.58  Seed = 19955
Figure C.59  Seed = 3062
Figure C.60  Seed = 64067
Figure C.61  Seed = 7548
Figure C.62  Seed = 7814
Figure C.63  Seed = 5114
Figure C.64  Seed = 14372
Figure C.65  Seed = 6058
Figure C.66  Seed = 12369
Figure C.67  Seed = 11927
Figure C.68   Seed = 21116
Figure C.69  Seed = 18203
Figure C.70  Seed = 169400
Figure C.71  Seed = 176540
Figure C.72  Seed = 22041
Figure C.73  Seed = 13749
Figure C.74  Seed = 210080
Figure C.75  Seed = 22219
Figure C.76  Seed = 74890
Bibliography


