



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**MULTIVARIATE PROBABILITY DENSITY
ESTIMATION USING PIECEWISE AFFINE FUNCTIONS**

by

Gabriel M. Samudio

June 2019

Thesis Advisor:
Second Reader:

Johannes O. Royset
Robert A. Koyak

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2019	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE MULTIVARIATE PROBABILITY DENSITY ESTIMATION USING PIECEWISE AFFINE FUNCTIONS			5. FUNDING NUMBERS	
6. AUTHOR(S) Gabriel M. Samudio				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) TRAC-Monterey, Monterey, CA 93940			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) Continuous multivariate data is ubiquitous in U.S. Military Operations Research. Since continuous distributions are fully characterized by their probability density functions, we concentrate on estimating such functions. Current estimation methods perform well for low dimensions; however, they can be too restrictive to capture the actual data characteristics, and can become intractable beyond three dimensions. This work develops a new estimation technique that seeks to increase flexibility and mitigate the curse of dimensionality. We achieve both by modeling the actual density using piecewise affine functions; however, we impose a nonconvex maximum likelihood optimization problem. The problem includes nine parameters, which can each affect the resulting estimate likelihood value and computation time. We conduct case studies on estimating the density for data up to five dimensions on sample sizes as low as 100 points. The results indicate progress in moderating the nonconvexity challenge to optimize the likelihood, and demonstrate potential advantages over currently used methods.				
14. SUBJECT TERMS maximum likelihood, piecewise affine functions, smooth maximum likelihood problem, relative log likelihood			15. NUMBER OF PAGES 67	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**MULTIVARIATE PROBABILITY DENSITY ESTIMATION USING
PIECEWISE AFFINE FUNCTIONS**

Gabriel M. Samudio
Major, United States Army
BA, University of Notre Dame, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

**NAVAL POSTGRADUATE SCHOOL
June 2019**

Approved by: Johannes O. Royset
Advisor

Robert A. Koyak
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Continuous multivariate data is ubiquitous in U.S. Military Operations Research. Since continuous distributions are fully characterized by their probability density functions, we concentrate on estimating such functions. Current estimation methods perform well for low dimensions; however, they can be too restrictive to capture the actual data characteristics, and can become intractable beyond three dimensions. This work develops a new estimation technique that seeks to increase flexibility and mitigate the curse of dimensionality. We achieve both by modeling the actual density using piecewise affine functions; however, we impose a nonconvex maximum likelihood optimization problem. The problem includes nine parameters, which can each affect the resulting estimate likelihood value and computation time. We conduct case studies on estimating the density for data up to five dimensions on sample sizes as low as 100 points. The results indicate progress in moderating the nonconvexity challenge to optimize the likelihood, and demonstrate potential advantages over currently used methods.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1	Introduction	1
1.1	Estimation Methods	2
1.2	Thesis Organization	5
2	Methodology	7
2.1	Candidate Function Class	7
2.2	Smooth Maximum Likelihood Problem	10
2.3	Algorithm	12
3	Empirical Results	19
3.1	Initial Estimate Methodology	21
3.2	Parallel Processing.	22
3.3	Synthetic Sample Size	27
3.4	Combined Effects	29
3.5	Improvement Search Phase Experiments	37
3.6	Five-Dimension Experiments	42
4	Conclusions	45
	Appendix: Computation	47
	List of References	49
	Initial Distribution List	51

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

Figure 1.1	Blind Men and the Elephant	1
Figure 1.2	Epi-spline Approximation Example	4
Figure 2.1	Piecewise Affine Function Example	8
Figure 2.2	Smooth Approximation Example	10
Figure 3.1	Univariate and bivariate f^0 Plots	20
Figure 3.2	Univariate Results for Different Initial Estimate Techniques . . .	22
Figure 3.3	Univariate Results for Parallel and Non-parallel Processes	23
Figure 3.4	Univariate Parallel and Non-parallel Process Comparison	24
Figure 3.5	Bivariate Results for Parallel and Non-parallel Processes	25
Figure 3.6	Bivariate Parallel and Non-parallel Process Comparison	26
Figure 3.7	Univariate size of m Comparison	28
Figure 3.8	Bivariate size of m Comparison	29
Figure 3.9	Univariate Experiment Number of Components Comparison . . .	31
Figure 3.10	Univariate Experiment γ and λ Comparison	32
Figure 3.11	Bivariate Experiment Number of Pieces Comparison	33
Figure 3.12	Bivariate Experiment γ and λ Comparison	34
Figure 3.13	Three-Dimension Experiment Number of Pieces Comparison . .	35
Figure 3.14	Three-Dimension Experiment γ and λ Comparison	36
Figure 3.15	Univariate Improved Search Results	37
Figure 3.16	Univariate Improved Search Plot	38
Figure 3.17	Bivariate Improved Search Results	39

Figure 3.18 Bivariate Improved Search Plots 40

Figure 3.19 Three-Dimension Improved Search Results 41

Figure 3.20 Five-Dimension Improvement Search Results 43

List of Acronyms and Abbreviations

DoD	Department of Defense
NPS	Naval Postgraduate School
MLE	Maximum Likelihood Estimator
ML	Maximum Likelihood
IPOPT	Interior Point Optimizer software
SMLP	Smooth Maximum Likelihood Problem
RLL	Relative Log Likelihood
OLH	Orthogonal Latin Hypercube

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

We consider methods to estimate the multivariate distribution of a given sample, assumed to be derived from an unknown density function. Multivariate distribution estimation is a central first step to understanding data and quantifying the uncertainty associated with multiple variables. Current estimators perform well under certain circumstances; however, they may suffer from inflexibility and can become intractable in higher than three dimensions. Most parametric estimators are fast to compute, and can approximate the actual density very well, but they rely on restrictive data assumptions. Nonparametric methods can be less reliant on assumptions, but they require exponentially larger sample sizes as the dimension increases. Other methods employ the Maximum Likelihood principle and form estimates by optimizing over a specific class of candidate functions. These estimators have shown promise for certain classes, but they can under-perform if the actual density does not belong to the specified class.

This thesis develops a new multivariate density estimator that seeks to improve current method flexibility, and to mitigate the curse of dimensionality. Similar to current non-parametric estimators, our method also relies on the Maximum Likelihood principle. We optimize over a broad class of functions that virtually encompasses any probability density function, thereby improving flexibility. We use piecewise affine functions to model the actual density, which allows the optimization problem to grow linearly in the dimension. While this new estimator is both flexible and can scale to high dimensions, it imposes a nonlinear and nonconvex optimization problem. Problem solutions are therefore not guaranteed to be globally or locally optimal, as the nonlinear optimization only detects stationary points satisfying the Karush–Kuhn–Tucker conditions. This work develops an algorithm for obtaining solutions that yield increasing log likelihood values, thereby moderating the nonconvexity challenge in crafting a flexible estimate.

We consider case studies in one, two, three, and five dimensions, in which we implement our algorithm to estimate the distribution from a synthetic data set generated from a known density function. We acknowledge in practice we would not know the actual density, but for this pilot study we evaluate our new estimator by comparing the estimate log likelihood against the log likelihood achieved by the actual density. There are many metrics for judging

density estimates such as proximity to the actual density, or the χ^2 goodness of fit p-value. This thesis focuses solely on attaining the estimate that yields the highest log likelihood on a given sample.

The results demonstrate that our new estimator can achieve log likelihood values as high as the actual density up to two dimensions. We observe a slight degradation in the log likelihood achieved in three dimensions, and a larger degradation in five dimensions. The results indicate that certain parameters can improve the log likelihood, but can also prolong run times. Further analysis on different algorithmic implementations is required to continue to develop this estimator. We show that our approach offers advantages over widely used estimation methods, and is viable for estimating multivariate distributions up to five dimensions on sample sizes as low as 100 observations.

CHAPTER 1: Introduction

Multivariate probability distributions enable leaders to understand and glean insights from multiple interdependent factors, which is of paramount importance to the U.S. military. As of 2019, most rebel factions in Syria have some degree of affiliation with a terrorist organization. Department of Defense (DoD) leaders need a method for determining which rebel faction has an irregular degree of risk and should not be trusted for inclusion in military partnerships that involve training and equipping with lethal weapons. Finding anomalies to forecast risk and to identify potential opportunities relies on estimation of a multivariate probability distribution as means to quantify signals for anomalous factions. In general, leaders' decisions are severely debilitated when they do not have awareness of the interdependence between factors in a situation, or do not have a way of prioritizing multiple factors. Analogous to the "Blind Men and the Elephant" parable (Thompson 2013), where blind men characterize the entire elephant based on the small portion they can feel, leaders with limited awareness risk misrepresenting the situation and making ill-informed decisions; see Figure 1.1.



Figure 1.1. Illustrating the "Blind Men and the Elephant" parable. Source: (Thompson 2013).

Multivariate probability distribution estimation is a way for leaders to become "aware of the entire elephant." It is a central first step to understanding data and quantifying the uncertainty associated with multiple variables. Military leaders rely on analysts and planners to glean insights, measure uncertainty, and make data-supported recommendations. Strategic level

planners using computer-based simulation models study multivariate output to conduct analysis of alternatives, emerging technology validation, and to forecast future threats. These models often implement assumed or hypothesized probability distributions to govern automatic behavior in the model. Further applications of multivariate distributions in the military include imputing missing data, constructing predictive models, and identifying dependencies.

Analysts typically begin their data exploration by visualizing, quantifying, and categorizing the distribution. Is the data normally distributed, or does it follow a different parametric distribution, or is a nonparametric model more appropriate? Does it have multiple modes? What is the mean? What is the range of the data? What is the likelihood that I will encounter data as extreme or more extreme than a given value? How spread out is my data? Are there outliers? What is the probability of the best and worst outcomes? All these important questions are addressed by estimating the multivariate probability distribution.

In this thesis, we develop a novel methodology for estimating continuous multivariate probability distributions and thereby address situations where the data is assumed to vary over a continuum of possible values. Since continuous distributions are fully characterized by their probability density functions, we concentrate on estimating such functions. The methodology relies on the Maximum Likelihood (ML) principle, which is computed over a large class of candidate density functions that extend beyond the familiar parametric models.

1.1 Estimation Methods

Practitioners employ both parametric and nonparametric approaches in constructing multivariate density function estimates. Parametric methods have the advantage of rapid computation, however suffer from inflexibility because they may fail to capture the actual data characteristics. Some nonparametric methods also have the advantage of rapid computation in low dimensions, and are more flexible than parametric methods in capturing irregular nuances in the data; however, they suffer from the curse of dimensionality.

1.1.1 Parametric Methods

Parametric methods for estimating density functions center on assuming given data is distributed according to a parametric distribution. The resulting parameters, such as the

μ and σ^2 for an assumed Gaussian Distribution, or the α and β for an assumed Weibull Distribution, are often determined by likelihood maximization. Parametric methods are useful for rapid modelling, however are unable to estimate a wide range of multivariate density functions; see (Chang and Sangrey 2019) for a discussion on parametric sensitivity to assumption deviations.

1.1.2 Nonparametric Methods

Common nonparametric methods for estimating multivariate density functions perform well for low-dimensional problems, but are challenged in higher than three dimensions. The widely used Kernel density estimator, pioneered in (Fix and Hodges 1951), superpositions a kernel function on each observation within a bandwidth (Liu and Wong 2014). Kernel estimate accuracy is sensitive to the bandwidth and kernel type, therefore methods such as cross validation are required to select an appropriate bandwidth and kernel type, otherwise the estimate may fail to be consistent. Cross validation becomes increasingly difficulty in high dimensions due to the curse of dimensionality, where there is an exponentially growing need for data as the dimension increases and the difficult of selecting a reasonable bandwidth multiplies. When using the common Gaussian kernel density estimator, if you require the same accuracy as 50 observations in one dimension, you will need more than 10^6 observations in ten dimensions (Nagler and Czado 2016).

An alternative to Kernel methods is M-estimators, which construct estimates by solving the ML optimization problem

$$\max_{f \in F} \prod_{j=1}^n f(x^j), \quad (1.1)$$

where F is a class of candidate density functions and $x^1, \dots, x^n \in S \subset \mathbb{R}^d$ are sample points assumed to be generated by independent sampling from a random vector. Analysts employing M-estimators restrict F to a particular class of functions to bound (1.1), ensure an estimator exists, and hopefully make (1.1) computationally tractable. If F includes the unimodal densities, then (1.1) will be unbounded and the ML estimator fails to exist; see for example (Cule et al. 2009). The computational challenge is often nontrivial. If F contains general functions, (1.1) becomes an infinite-dimensional problem requiring optimization

over an infinite number of parameters. In restricting F to be a particular class of functions we can gain tractability, however the M-estimator's Kullback–Leibler divergence from the actual density may never reach zero if the actual density does not reside in F . Several choices for F have been shown to be tractable in the univariate case. Epi-splines are studied in (Royset and Wets 2015), the monotonic, monotonic and convex, U-shaped, and unimodal with known mode classes in (Sankaran and Unnikrishnan Nair 2009), the log-concave class in (Birke 2009), and the class of monotonic transformations of convex functions in (Seregin and Wellner 2010).

In higher dimensions, recent work has shown progress using the log-concave class in (Cule and Samworth 2010) and multivariate epi-splines in (Royset and Wets 2016) and (Royset and Wets 2017) to define F . The log-concave class is promising and somewhat computationally tractable, although it relies on an assumption that the density function is indeed log-concave. Epi-splines are more flexible, as they can approximate any function within arbitrary error by partitioning the domain of the function to be approximated by k subsets (Royset and Wets 2016) and (Royset and Wets 2017). As k increases, the approximation error decreases because finer subdivision allows the approximation another opportunity to make a turn in the direction of the actual function; see Figure 1.2.

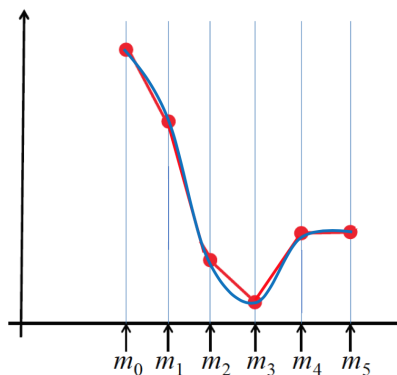


Figure 1.2. The blue function is the function to be approximated. The red epi-splines are able to produce a very close estimate with a partition consisting of five subsets.

The epi-spline estimation technique has the additional advantage of incorporating constraints such as convexity, log-convexity, continuity, smoothness, location of modes, and

approximate mean, which all work to limit the class of functions F and yield more accurate estimates. The first application of a three-dimensional exponential epi-spline probability density estimate was used in 2018 by LTC Michael Teter to quantify risk associated with U.S. Army future technology investments. Only a small number of subject matter expert elicitation inputs were available to inform decision makers on the development risk and potential opportunity of a number of investment options. Exploiting the epi-spline technique flexibility enhanced decision maker's risk understanding and provided a more conservative risk assessment than would have otherwise been available using parametric estimation means; see (Teter et al. 2016) for further details on this epi-spline application and its advantages. The drawback to the epi-spline method is that it suffers from the curse of dimensionality. Since the number of sets in the partition increases exponentially as the dimension increases, computing the epi-spline estimate beyond three dimensions quickly requires advanced computational resources.

1.2 Thesis Organization

Chapter 2 describes our new methodology for estimating a multivariate probability distribution. We formulate a nonlinear program and identify the associated challenges. We then propose techniques for meeting these new challenges and formalize our method in an algorithm.

Chapter 3 tests our new density estimation method. We show that it is viable for estimating densities up to five dimensions, and explore the performance under various inputs. We demonstrate an approach for determining ideal formulation parameter values and show the effects on the estimate quality and run time as the dimension and sample size increase.

Chapter 4 concludes by summarizing the advantages and limitations in using our method, and recommends future developmental research to the community of interest.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2: Methodology

We develop a Maximum Likelihood Estimator (MLE) for a multivariate probability density function $f^0 : S \subset R^d \rightarrow [0, \infty)$ using a sample $X = \{x^1, \dots, x^n\} \subset S$ generated by independent sampling from f^0 . Specifically, the estimator f^n is an optimal solution of (1.1) or, equivalently,

$$f^n \in \operatorname{argmin}_{f \in F} \frac{-1}{n} \sum_{j=1}^n \log f(x^j), \quad (2.1)$$

where F is a class of density functions. As discussed in Chapter 1, if F consists of only log-concave densities, monotone densities, convex densities, or certain other special classes, then f^n is well understood and maybe even fully characterized; see (Cule and Samworth 2010), (Sankaran and Unnikrishnan Nair 2009), (Birke 2009), and (Seregin and Wellner 2010). When F consists of splines or epi-splines, f^n can again be determined; see (Royset and Wets 2015), (Royset and Wets 2016), and (Royset and Wets 2017) and the references therein. Although flexible, approaches based on splines or epi-splines rely on a mesh, i.e., a partition of S into a number of subsets, which becomes intractable for $d > 3$. This thesis is the first attempt to solve (2.1) for a general class of densities in a manner that has the potential to scale to higher dimensions. In particular, the approach results in an optimization problem that scales linearly in d . Thus, we avoid a mesh, which would have implied an exponential grow in problem size as d increases. Unfortunately, our approach comes with its own challenges: the resulting optimization problem is nonconvex. This chapter details our progress in meeting these challenges and solving (2.1).

2.1 Candidate Function Class

The class of upper-semicontinuous functions contains essentially all probability density functions, which can be approximated by piecewise affine functions of the difference-of-max type to an arbitrary level of accuracy as the number of pieces grow (Royset 2017, Theorem 3.1). Thus, by setting F to be the class of such piecewise affine functions we can

estimate any density function without being concerned about correct model specification, at least if the number of pieces is increased appropriately. In this thesis, we focus exclusively on this class, which can be expressed by piecewise affine functions of the form

$$f(x) = \max_{k=1,\dots,q} \{\langle a^k, x \rangle + \alpha_k\} - \max_{l=1,\dots,r} \{\langle b^l, x \rangle + \beta_l\} \quad \forall x \in S. \quad (2.2)$$

The function value at a point x is the difference of the maximum values of each collection of affine functions evaluated at x . Our convention uses q and r to designate the number of affine functions in the first and second collection respectively, and defines a^k and b^l as d -dimensional vectors, and α_k and β_l as scalars. Figure 2.1 shows an example of a piecewise affine function in one dimension.

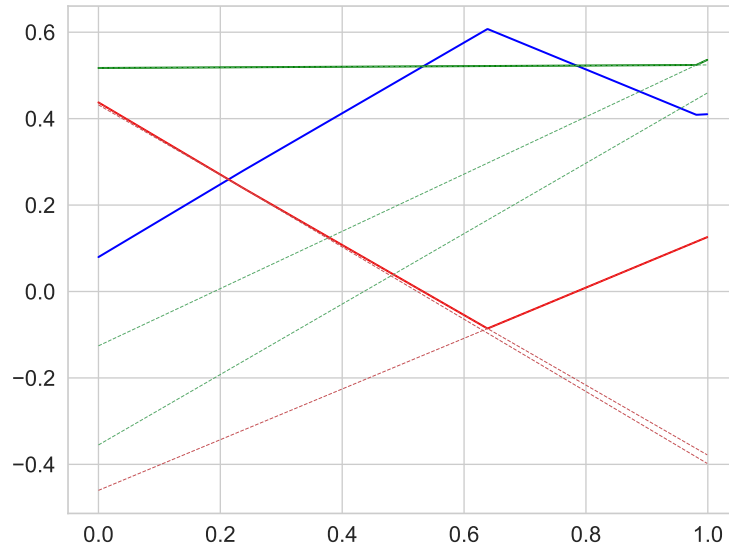


Figure 2.1. The blue line is the difference of the maximum values of the green lines minus the red lines. The bold red and green lines are the maximum of their respective component affine functions, shown as the dashed lines. The plot illustrates the flexibility of piecewise affine functions.

2.1.1 Smooth Approximation

The maximum operators in a piecewise affine function prevent us from using gradient-based optimization, thus we approximate f with

$$f_\gamma(x) = \frac{1}{\gamma} \log \sum_{k=1}^q e^{\gamma(\langle a^k, x \rangle + \alpha_k)} - \frac{1}{\gamma} \log \sum_{l=1}^r e^{\gamma(\langle b^l, x \rangle + \beta_l)}, \quad (2.3)$$

which is differentiable and can become arbitrarily close to f as γ grows (Polak et al. 2003). We have the inequalities,

$$0 \leq \frac{1}{\gamma} \log \sum_{k=1}^q e^{\gamma(\langle a^k, x \rangle + \alpha_k)} - \max_{k=1, \dots, q} \{\langle a^k, x \rangle + \alpha_k\} \leq \frac{\log q}{\gamma} \quad (2.4)$$

and

$$0 \leq \frac{1}{\gamma} \log \sum_{l=1}^r e^{\gamma(\langle b^l, x \rangle + \beta_l)} - \max_{l=1, \dots, r} \{\langle b^l, x \rangle + \beta_l\} \leq \frac{\log r}{\gamma}, \quad (2.5)$$

which together show that

$$\sup_{x \in \mathbb{R}^d} |f_\gamma(x) - f(x)| \leq \frac{\log \max\{q, r\}}{\gamma}. \quad (2.6)$$

Since we control γ and, largely, also q and r , the smoothing error can be made arbitrarily small. Figure 2.2 illustrates this smooth approximation.

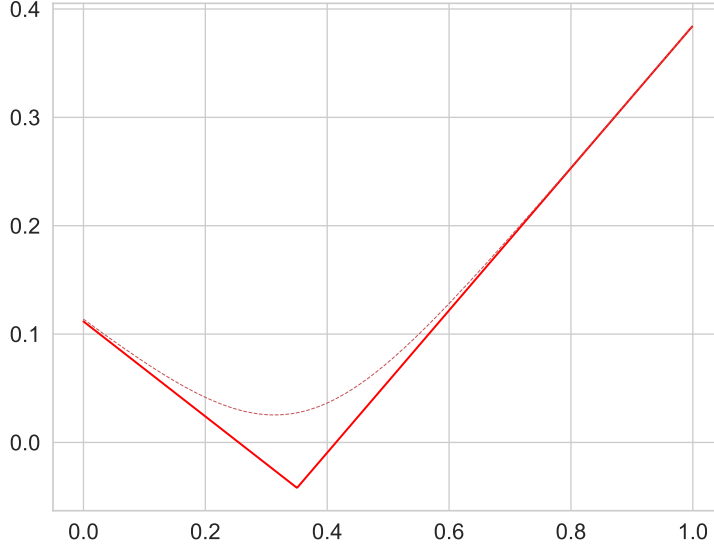


Figure 2.2. The bold line is a piecewise affine function and the dashed line is its smooth approximation.

We define F_γ as the set of smooth functions, f_γ , and optimize over the coefficient values a^k , b^l , α_k and β_l .

2.2 Smooth Maximum Likelihood Problem

We need to enforce non-negativity and integration to one on (2.1) to ensure the solution is a valid density function. The integral quickly becomes difficult to compute directly in high-dimensions. We enhance tractability by generating a synthetic sample $\Xi = \{\xi^1, \dots, \xi^m\} \subset S$ from a known density φ to perform the following approximation:

$$\int f_\gamma(x) dx = \int \frac{f_\gamma(x)}{\varphi(x)} \varphi(x) dx \approx \frac{1}{m} \sum_{j=1}^m \frac{f_\gamma(\xi^j)}{\varphi(\xi^j)}. \quad (2.7)$$

We further enhance tractability by moving the integral constraint to the objective function, which still enforces integration to one; see (Silverman 1982). These formulation modifications lead to the following nonlinear program, which we define as the Smooth Maximum

Likelihood Problem (SMLP), and present it using the Naval Postgraduate School (NPS) format from (Brown and Dell 2007).

Parameters

$x^1, \dots, x^n \in S \subset \mathbb{R}^d$	Data
$\xi^1, \dots, \xi^m \in S \subset \mathbb{R}^d$	Synthetic Data
ε	Positive threshold
γ	Smooth Approximation parameter
q	Number of affine functions in the first collection
r	Number of affine functions in the second collection
λ	Regularization parameter

Decision Variables

$$a^k \in \mathbb{R}^d, \alpha_k \in \mathbb{R}, b^l \in \mathbb{R}^d, \beta_l \in \mathbb{R} \quad \text{for } k = 1, \dots, q \text{ and } l = 1, \dots, r$$

Formulation

$$\begin{aligned}
\text{SMLP: } \min_{f_\gamma \in F_\gamma} & \quad \frac{-1}{n} \sum_{j=1}^n \log f_\gamma(x^j) + \pi(f_\gamma) + \frac{1}{m} \sum_{j=1}^m \frac{f_\gamma(\xi^j)}{\varphi(\xi^j)} \\
\text{s.t.} \quad & \quad f_\gamma(\xi^j) \geq 0 \quad \forall j = 1, \dots, m \\
& \quad f_\gamma(x^j) \geq \varepsilon \quad \forall j = 1, \dots, n
\end{aligned}$$

Discussion

We include the regularization,

$$\pi(f_\gamma) = \lambda \left(\sum_{i=1}^q \|a^i\|^2 + \sum_{i=1}^r \|b^i\|^2 \right), \quad \lambda \geq 0 \quad (2.8)$$

in the objective function for numeric stability and to prevent overfitting by penalizing the magnitude of the slope coefficients. We set $\varepsilon > 0$ to enforce that f_γ is positive over $\{x^1, \dots, x^n\}$, thereby guarding the objective function from becoming undefined.

The SMLP is both nonconvex and possibly ill-conditioned. Nonconvexity holds for any q and r . The regularization function π is always convex, however the integral approximation is only convex if $r = 1$. The first term is not guaranteed to be convex when $r = 1$ since it is a sum of composition functions, where the outside function, $\frac{-1}{n} \log$, is decreasing, and the inside function, f_γ , is not necessarily concave (Royset 2019). If $q = 1$ the first term is convex, but the integral becomes nonconvex. Exacerbating the computational challenge, certain values of the parameters can cause ill-conditioning, especially large values of gamma; see (Polak et al. 2003) for a discussion. The following section formalizes our method to tackle these challenges.

2.3 Algorithm

Our procedure attempts to find the coefficients that comprise (2.2), $\{a^k, \alpha_k, b^l, \beta_l\}$ for $k = 1, \dots, q$ and $l = 1, \dots, r$, that yield the highest log likelihood within a specified time. The general work-flow alternates between finding local solutions, and generating start points from prior solutions. The algorithm consists of two phases: the Initial Estimate, and the Improvement Search Phase. The Initial Estimate Phase obtains a local solution from random input, which becomes the start point that launches the Improvement Search Phase. During the Improvement Search Phase we obtain new local solutions using previously optimized coefficients as start points. The algorithm concludes by terminating the search for improved solutions, and converting the last best solution to the difference-of-max form (2.2), which forms the probability density estimate.

SMLP Algorithm**Output:** f^n **Input:** X

Declare Variables: $p = \{a^k, \alpha_k, b^l, \beta_l\}$ for $k = 1, \dots, q$ and $l = 1, \dots, r$ p_b = best solution p ; p_n = new solution p bll = best log likelihood; nll = new log likelihood**Initialize:** $\Xi, p, \gamma, \varepsilon, \lambda, \mathcal{V}, v, \Omega$ **Initial Estimate Phase:** $f_\gamma \leftarrow$ Start Point Algorithm with input: $\{X, \Xi, p, \gamma, \varepsilon, \mathcal{V}, v\}$ $p_b \leftarrow$ Solve SMLP with input: $\{f_\gamma, X, \Xi, \varepsilon, \lambda\}$ $f \leftarrow$ Conversion Algorithm with input: $\{X, \Omega, p_b, \varepsilon\}$ $bll \leftarrow \sum_{j=1}^n \log f(x^j)$ **Improvement Search Phase:****while** *Termination Conditions Not Satisfied* **do** **if** *Termination Conditions Satisfied* **then** $f \leftarrow$ Conversion Algorithm with input: $\{X, \Omega, p_b, \varepsilon\}$ **return** f **else** $p_b[i] \leftarrow p_b[i] + \theta$ where $\theta \sim \mathcal{N}(0, |\eta p_b[i]|)$ for i in range(len(p_b)) $f_\gamma \leftarrow$ Start Point Algorithm with input: $\{X, \Xi, p, \gamma, \varepsilon, \mathcal{V}, v\}$ $p_n \leftarrow$ Solve SMLP with input: $\{f_\gamma, X, \Xi, \varepsilon, \lambda\}$ $f \leftarrow$ Conversion Algorithm with input: $\{X, \Omega, p_n, \varepsilon\}$ $nll \leftarrow \sum_{j=1}^n \log f(x^j)$ **if** $nll > bll$ **then** $bll \leftarrow nll$ $p_b \leftarrow p_n$ **else**

continue

end **end****end**

Initial Estimate Phase

This phase starts by setting initial values for m , ε , γ , q , r , and λ (The initializations of $\mathcal{V}$, ν and Ω are covered in the Conversion Algorithm section). We define p as the vector of coefficients, $\{a^k, \alpha_k, b^l, \beta_l\}$ for $k = 1, \dots, q$ and $l = 1, \dots, r$, and use p_b , p_n , bll , and nll for tracking the best and new solution and log likelihood values. We generate a random start point p , with q and r affine functions in the first and second collection respectively. In our research, we randomize the initial p with values uniformly distributed between -0.001 and 0.001 . We then sample m points from S distributed according to a known density function φ , and refer to this sample as Ξ . In our research, we set φ as the uniform distribution parametrized by the bounds on S in each dimension. Starting with all the coefficients near zero produces a function near the zero function, which does not integrate to one. The Start Point Algorithm adjusts p and γ so that f_γ is positive and integrates to one over Ξ . We then attempt to solve this SMLP instance defined by the preceding parameters by employing a nonlinear solver. The SMLP is a smooth nonlinear program, consisting of $m + n$ constraints and $(d + 1)(q + r)$ decision variables, so the problem size can grow linearly in the dimension d . The problem becomes increasingly ill-conditioned as γ grows, which causes nonlinear programming solvers to degrade in performance; see (Polak et al. 2003) for a discussion. Each solution is only a stationary point, satisfying the Karush–Kuhn–Tucker conditions, and not necessarily a local minimum or an improvement from a previous solution. If the nonlinear program solver cannot find a solution, then we decrease γ to guard against encountering a number larger than the range of values able to be represented by double precision floating point numbers, and increase q and r to improve flexibility in minimizing the objective function under the SMLP constraints. We set the initial solution as the current best solution, p_b , since it is the best solution encountered so far. The Conversion Algorithm constructs a difference-of-max function, f , from the solution coefficients, p_b , and makes additional fine tuning adjustments. We then conclude this phase by computing the log likelihood of f , and establishing this value as the current best log likelihood, bll .

Improvement Search Phase

The termination conditions for the Improvement Search Phase can be a specified computation time, number of solution attempts, number of improvements, and/or a certain objective function value. This phase advances toward a global optimal solution by traversing the SMLP feasible region using improved solutions as launching points. We shift the current

best solution coefficients, p_b , by adding a random variable θ , that is distributed according to the Normal Distribution, parametrized by $\mu = 0$ and $\sigma = |\eta \times \text{current coefficient value}|$. In our research, we set $\eta = 5$, and increase it by 50% if we do not find an improvement after ten iterations. If we find ten consecutive infeasible solutions, then we reduce η by 10%. The Start Point Algorithm then adjusts these newly shifted coefficient values to ensure strict feasibility. We obtain a new solution from this problem instance, then the Conversion Algorithm constructs the difference-of-max form function, f , from the new solution coefficients, p_n . We then compute the log likelihood, nll , and compare it to the highest log likelihood computed thus far in the algorithm, bll . If $nll > bll$, then we have found an improved solution. This value nll becomes bll , and p_n becomes p_b . If we did not find an improved solution, then we do not change the current p_b and bll . We repeat the previous steps until reaching termination conditions.

We now describe the two subroutines that makeup the SMLP algorithm, Start Point and Conversion, and discuss our implementation.

2.3.1 Start Point Algorithm

The starting function, f_γ , for each problem instance is composed of coefficients p and the smooth approximation parameter γ . Recall from (2.7) that as γ increases the approximation error decreases, so we are inclined to make γ arbitrarily large. However, γ can cause the solver to encounter values larger than can be represented by double precision floating numbers since it is a multiplier for the power of the exponents in (2.4). The following algorithm consists of one subroutine, the Max Smoothing Algorithm, which finds an approximately maximum γ that prevents encountering such large values. The Start Point Algorithm then constructs f_γ from this maximum feasible γ , and enforces strict non-negativity and integration to one over Ξ .

Start Point Algorithm**Output:** f_γ **Input:** $X, \Xi, p, \gamma, \varepsilon, \mathcal{V}, \nu$

Declare Variables: $v = \{f_\gamma(x^1), \dots, f_\gamma(x^n), f_\gamma(\xi^1), \dots, f_\gamma(\xi^m)\}$ **Initialize:** f_γ **Enforce Non-negativity:** $v, \gamma \leftarrow$ Max Smoothing Algorithm with input: $\{X, \Xi, f_\gamma, \mathcal{V}, \nu\}$ **while** $-\min v > \varepsilon$ **do** $\alpha_k \leftarrow \alpha_k + \max\{-\min v, 0\} + \varepsilon \quad \forall k = 1, \dots, q$
 $v, \gamma \leftarrow$ Max Smoothing Algorithm with input: $\{X, \Xi, f_\gamma, \mathcal{V}, \nu\}$ **end****Enforce Integration to One:**

$$I \leftarrow \frac{1}{m} \sum_{j=1}^m \frac{f_\gamma(\xi^j)}{\varphi(\xi^j)}$$

$$a^k \leftarrow \frac{a^k}{I}, \alpha_k \leftarrow \frac{\alpha_k}{I}, b^l \leftarrow \frac{b^l}{I}, \beta_l \leftarrow \frac{\beta_l}{I} \quad \forall k = 1, \dots, q \text{ and } \forall l = 1, \dots, r$$

return f_γ

Overview

We define v as the set of f_γ evaluations on every $x \in X$ and every $\xi \in \Xi$. We construct f_γ from the initialized γ , and initial random coefficients p . The Max Smoothing Algorithm produces v and a maximum feasible γ . This algorithm begins by trying to evaluate an initial f_γ on each $x \in X$ and each $\xi \in \Xi$. If we do not encounter a larger number than can be represented by double precision floating numbers we increase γ by a factor of $\mathcal{V}$. We then construct a new f_γ from the increased γ and try evaluating f_γ on every point again. We continue increasing γ until we encounter such a large number, which identifies a γ upper limit. We then gradually decrease γ by a factor of ν (which must be less than $\mathcal{V}$). In our research, we set $\mathcal{V} = 2$ and $\nu = 0.95$. We expect to continue to encounter these large numbers until our smoothing parameter γ is sufficiently small. Each time we exceed the threshold for representable numbers we decrease γ and try evaluating again. The first γ to render successful evaluation on every point is the approximately maximum feasible γ . This procedure is formalized below.

Max Smoothing Algorithm**Output:** v, γ **Input:** $X, \Xi, f_\gamma, \mathcal{V}, \nu$

try: $v \leftarrow f_\gamma$ evaluated $\forall x^1, \dots, x^n$ and $\forall \xi^1, \dots, \xi^m$ $\gamma = \mathcal{V}\gamma$ **catch Overflow Error:****while** *Not Feasible* **do****try:** $\gamma = \nu\gamma$ $v \leftarrow f_\gamma$ evaluated $\forall x^1, \dots, x^n$ and $\forall \xi^1, \dots, \xi^m$ **return** v, γ **catch Overflow Error:**

| continue

end**end****end**

After employing the Max Smoothing Algorithm, we next ensure that $f_\gamma(x) \geq \varepsilon$ for any x in X . If $\min v$ is negative, we increase f_γ by $-\min v + \varepsilon$. If $\min v$ is positive, but below ε , we increase f_γ by ε to move our start point well within the feasible region. Finally, we ensure integration to one by dividing every coefficient in p by the integral approximation, $\frac{1}{m} \sum_{j=1}^m \frac{f_\gamma(\xi^j)}{\varphi(\xi^j)}$.

2.3.2 Conversion Algorithm

This procedure occurs intermittently during the overall work-flow to identify improved solutions, and occurs at the conclusion to construct a final estimate from the best found solution. This conversion is formalized below.

Conversion Algorithm**Output:** f **Input:** $X, \Omega, p, \varepsilon$

Declare Variables: $u = \{f(x^1), \dots, f(x^n), f(\omega^1), \dots, f(\omega^M)\}$ **Initialize:** f, u **Enforce Non-negativity:****while** $-\min u > \varepsilon$ **do** $\alpha_k \leftarrow \alpha_k + \max\{-\min u, 0\} + \varepsilon \quad \forall k = 1, \dots, q$ Reconstruct f **end****Enforce Integration to One:** $I \leftarrow \frac{1}{M} \sum_{j=1}^M \frac{f(\omega^j)}{\varphi(\omega^j)}$ $f \leftarrow \frac{f}{I}$ **return** f

Overview

Recall that the SMLP constraints enforce non-negativity and integration to one over Ξ , which is a subset of S , the support for the density function we endeavor to estimate, f^0 . The Conversion Algorithm moves the estimate closer to non-negativity and integration to one over S by making adjustments on a larger subset of S of size M , which is distributed according to φ . We refer to this larger subset as Ω . In our research, we set $M = 5n$. The algorithm begins by constructing a difference-of-max form function from the input coefficients, p . We then define the variable u as the set of f evaluations on every $x \in X$ and every $\omega \in \Omega$. Similar to the Start Point Algorithm, we adjust each scalar in the first collection of affine functions by adding the appropriate amount to ensure f is at least ε above zero. We acknowledge that a density estimate is permitted to be zero; however, we enforce that f is positive so we can compute the log likelihood. Finally, we ensure integration to one over Ω by dividing every coefficient by the integral approximation, $\frac{1}{M} \sum_{j=1}^M \frac{f(\omega^j)}{\varphi(\omega^j)}$. While this is only an approximate integral over S , we can always improve accuracy by increasing M .

CHAPTER 3: Empirical Results

The methodology described in Chapter Two includes several inputs (such as the number of affine functions q and r , the synthetic sample size m , the smooth approximation parameter γ , and the regularization parameter λ), needed to formulate and solve the SMLP. This chapter tests the effects of these inputs by implementing them to estimate the density for data generated from a known density, f^0 . We begin by focusing on the Initial Estimate Phase to study the parameter effects on the solution quality and problem run time. We study the benefits of two different methods for generating an initial estimate. We then adopt the superior method, and proceed to comparing two programming methods. Next, we study the effects of the sample size, m , of the data Ξ (used to approximate the estimate integral). We conclude the Initial Estimate Phase study by conducting a designed experiment in one, two, and three dimensions to explore the interdependence between λ , γ , and the number of components q and r , to identify promising trends in parameter value ranges. We then progress to studying the Improvement Search Phase of the SMLP Algorithm. We conduct experiments in one, two, and three dimensions on 100-point sample sizes to measure the effects of increasing the dimension. We then vary the sample size and conduct two experiments in five dimensions to measure the effect of increasing the sample size.

The test density for this chapter is a mixture of three uniform densities,

$$f^0(x) = \begin{cases} 0.1 & -2 \leq x^j \leq -1 \quad \forall j = 1, \dots, d \\ \frac{0.4}{2^d} & 0.5 \leq x^j \leq 2.5 \quad \forall j = 1, \dots, d \\ \frac{0.5}{6^d - (1 + 2^d)} & \text{otherwise,} \end{cases} \quad (3.1)$$

where the support S ranges from -3 to 3 in each dimension. Figure 3.1 plots f^0 in one and two dimensions.

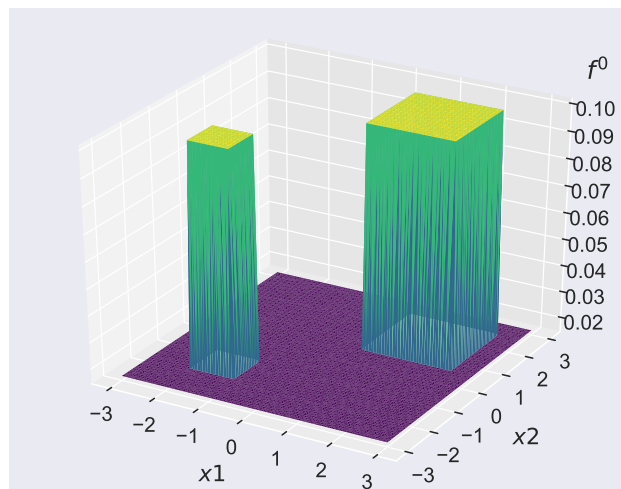
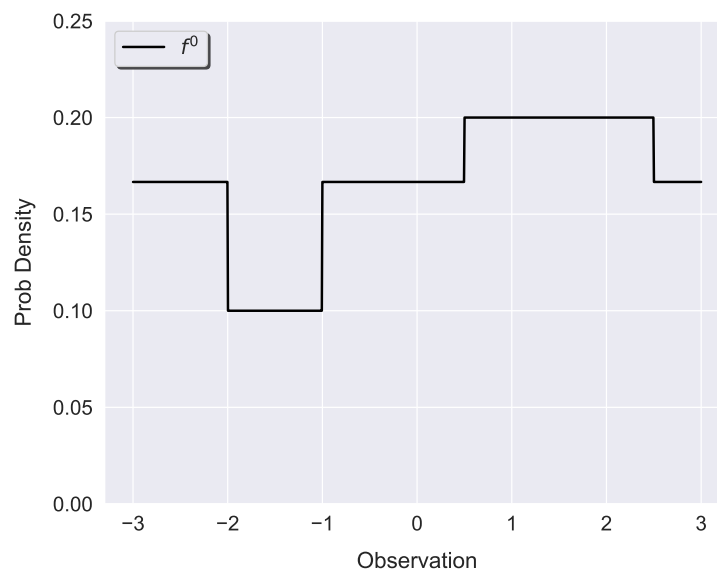


Figure 3.1. True Density f^0 in one and two dimensions.

We measure the SMLP solution quality by computing

$$\frac{\frac{1}{n} \sum_{j=1}^n \log f^0(x^j) - \frac{1}{n} \sum_{j=1}^n \log f^n(x^j)}{|\frac{1}{n} \sum_{j=1}^n \log f^0(x^j)|}, \quad (3.2)$$

which we define as the Relative Log Likelihood (RLL). This metric is the proportion of the actual density log likelihood that is different from the estimate log likelihood. Higher RLL values indicate better solutions, as solving the SMLP seeks to maximize $\frac{1}{n} \sum_{j=1}^n \log f^n(x^j)$. We acknowledge that probability density estimates are measured according to different metrics such as the proximity to the actual density, or the χ^2 Goodness of Fit p-value. This work focuses solely on attaining the estimate that yields the highest log likelihood on a given sample.

3.1 Initial Estimate Methodology

For this study, we develop two approaches for the Initial Estimate Phase, "Build Up" and "All at Once." In the first approach we reduce r to one, set q to the intended value, and then execute the Initial Estimate Phase for a preliminary solution. We then increase r by one and execute the Initial Estimate Phase again, starting from this preliminary solution. We repeat this process until we have an initial solution for the desired number of q and r . The second approach is simply solving for all q and r at once, instead of one at a time. We suspect there may be a computational advantage in the "Build Up" method because we are solving a sequence of small scale SMLP instances, which may cumulatively take less run time than solving the full problem at once. We compare these methods by implementing them to estimate the density of 100 different random f^0 samples of 1000 points. We set $\gamma = 50$, $m = 100$, q , $r = 5$, and $\lambda = \frac{1.0 \times 10^{-10}}{(q+r)d}$ for each experiment.

After 100 trials of both methods, we produce results favoring the method that solves for all q and r affine functions at once. Figure 3.2 shows the RLL and run time distribution for each method.

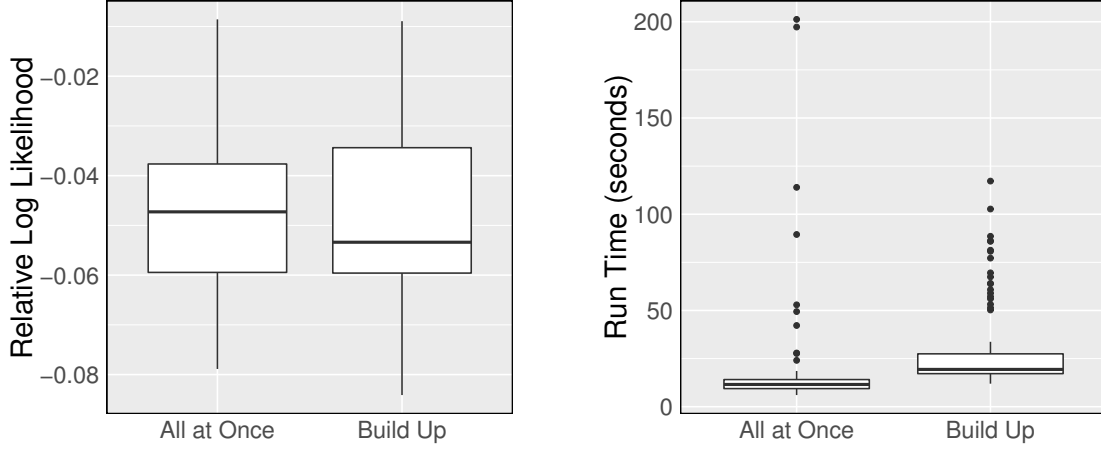


Figure 3.2. The two methods are largely similar for both RLL and run time, indicating little merit in breaking the SMLP into smaller scale problems.

The "All at Once" method has a slightly higher median RLL and slightly less variability. While this is not overwhelming evidence in favor of the "All at Once" method, we conclude there is not a clear advantage in breaking the SMLP into smaller problems for the Initial Estimate Phase. We adopt the "All at Once" method based on this evidence.

3.2 Parallel Processing

Multi-core processing can reduce computation time by executing multiple jobs in parallel. Unfortunately, one SMLP instance will not likely benefit from parallel processing because most nonlinear optimization relies on a single computation sequence. However, we can implement parallel processing by dividing S into K areas, where K is the number of core processors available for parallel execution. We then launch K SMLP instances in parallel from the sample data residing in each area. Each SMLP instance uses the same Ξ points for the integral estimate in the objective function, ensuring each area estimate integrates to one over S . The final estimate is the weighted sum,

$$f^n = \sum_{k=1}^K \psi_k f_k^n, \quad (3.3)$$

where ψ_k is the proportion of data, x^j , residing in area k . We test the benefits of parallel processing by executing parallel and non-parallel versions of the Initial Estimate Phase on 100 different random samples from f^0 of 100 points in one and two dimensions. We set $\gamma = 50$, $m = 100$, $q, r = 5$, and $\lambda = \frac{1}{(q+r)d}$ for each experiment.

Figure 3.3 compares the RLL and run time distribution for each process. The run times are only from the start of the SMLP Algorithm until the conclusion of the Initial Estimate Phase. These boxplots do not show strong evidence in favor of parallel processing, as the RLL distributions are very similar, with the non-parallel procedure showing a slightly higher median value. The parallel process run times appear significantly greater, however there is not a practical difference because the scale is in seconds.

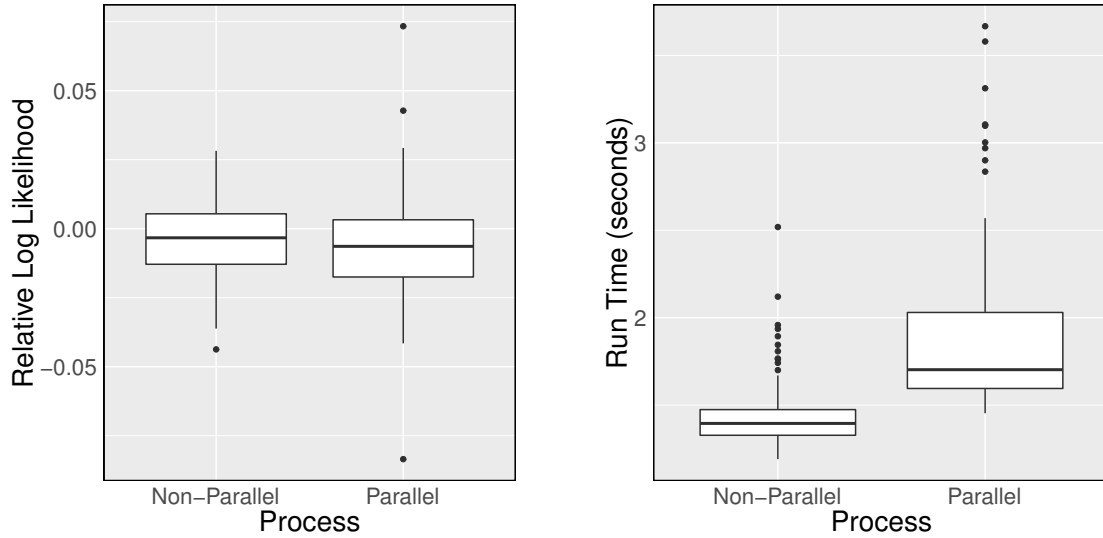
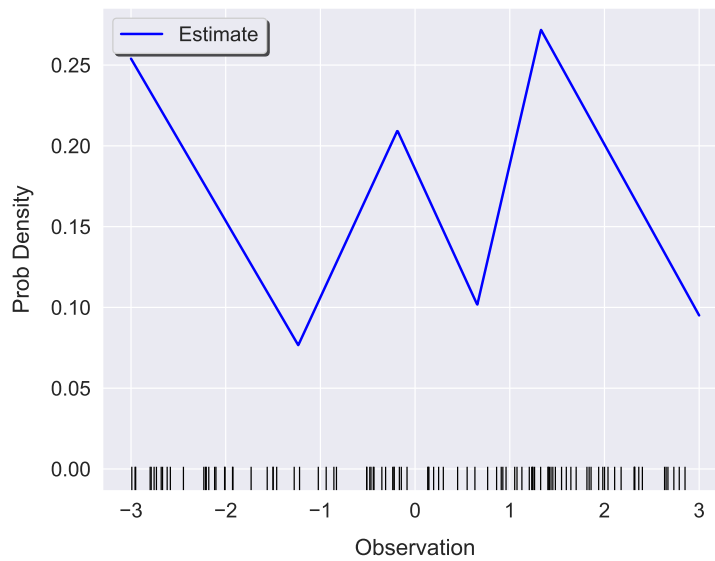
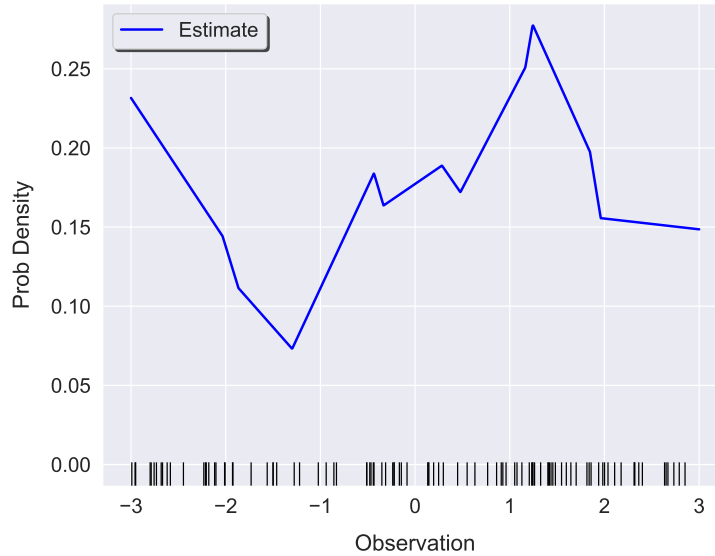


Figure 3.3. These boxplots show that parallel processing benefits are not substantial during the Initial Estimate Phase for univariate data.

We investigate further by plotting the solution with the median RLL for each version in Figure 3.5.



Non-Parallel Process Estimate



Parallel Process Estimate

Figure 3.4. The parallel process estimate consists of four density estimates, each composed of ten piecewise affine functions. The combined estimate then consists of forty piecewise affine functions, which produces the improved flexibility shown in the graph.

In contrast to the modest RLL and run time differences exhibited in the Figure 3.4 boxplots, Figure 3.5 shows a distinct difference between the two versions. The parallel version results in a clear flexibility advantage over the non-parallel version, as the parallel estimate has multiple peaks and valleys corresponding to high and low densities in the sample.

The results from the bivariate experiment show slight evidence against the parallel process. Figure 3.6 shows the non-parallel RLL distribution is slightly higher than the parallel version, and takes slightly less run time on average. However, this is not overwhelming evidence in favor of the non-parallel version, as the run times are within fractions of a second and the RLL distributions are largely nested.

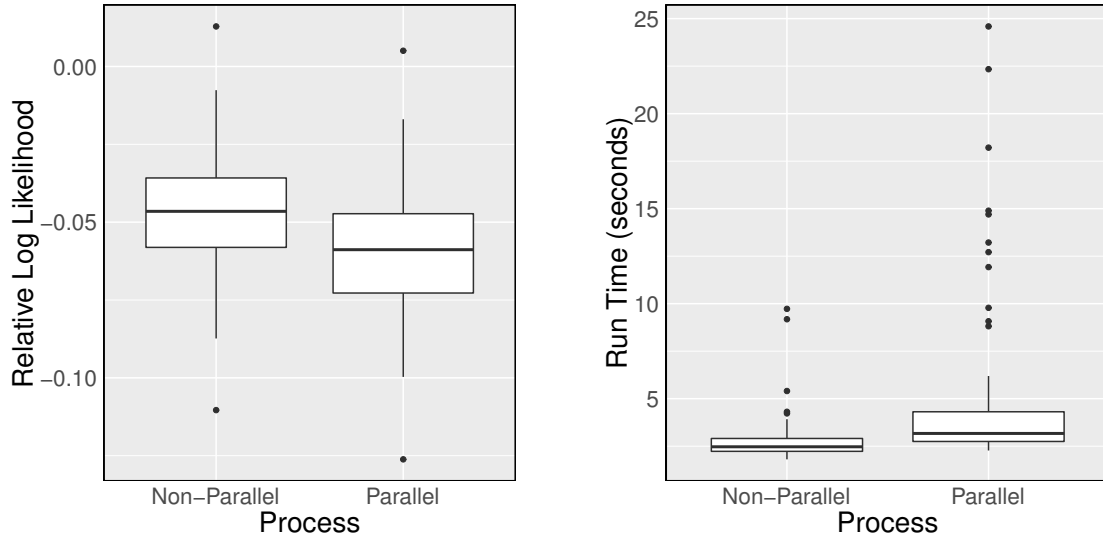
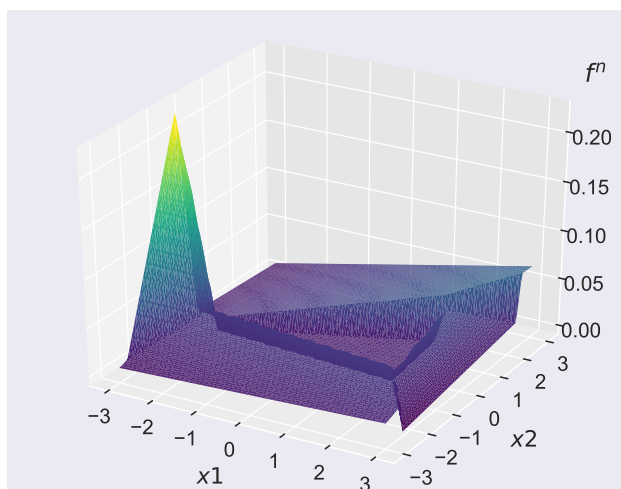
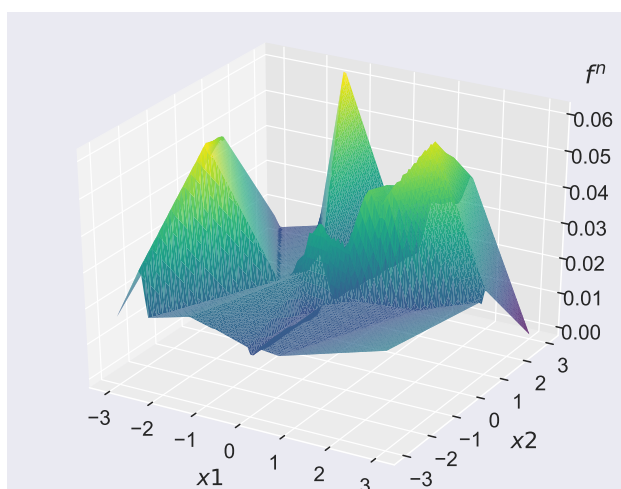


Figure 3.5. The parallel process incorporates four times as many piecewise affine functions into the estimate as the non-parallel estimate in the same amount of time.

We investigate the two processes visually by plotting the solution surface with the median RLL for each version in Figure 3.7.



Non-Parallel Process Estimate



Parallel Process Estimate

Figure 3.6. Similar to the univariate result, the additional piecewise affine functions incorporated into the parallel process estimate yield additional flexibility.

Similar to the one-dimension result, the parallel estimate signals high densities throughout S , whereas the non-parallel estimate only signals high density in one location.

While the results of this experiment are not conclusive, we adopt the parallel process because of the benefit of using more piecewise affine functions, which improves the estimate's ability to vary and capture effects.

3.3 Synthetic Sample Size

The number of synthetic points, m , affects the integral approximation precision and the number of constraints ($m + n$). We explore this parameter by conducting 100 Initial Estimate Phases on different random samples from f^0 in one and two dimensions. We set $\gamma = 50$, $q, r = 5$, and $\lambda = \frac{1.0 \times 10^{-10}}{(q+r)d}$ for each experiment. We conduct the univariate experiment on samples of size $n = 1000$, as a larger sample size will likely extract key differences in the results of various m values. We decrease the sample size to $n = 225$ for the bivariate experiment in order to procure rapid results.

Figure 3.8 suggests that increasing m can improve the RLL. The additional red dots in the boxplots are the sample means, which we include in this analysis to highlight the difference with the median values. The left boxplot shows each sample mean RLL is slightly lower than each median, indicating left skewed distributions. Figure 3.8 also shows that the run time generally increases as m increases. The sample mean run times for $m = 100$ and $m = 250$ are higher than the medians, caused by large run time outliers. This evidence shows that the advantage in RLL of a high m value does not necessarily cause run times to increase.

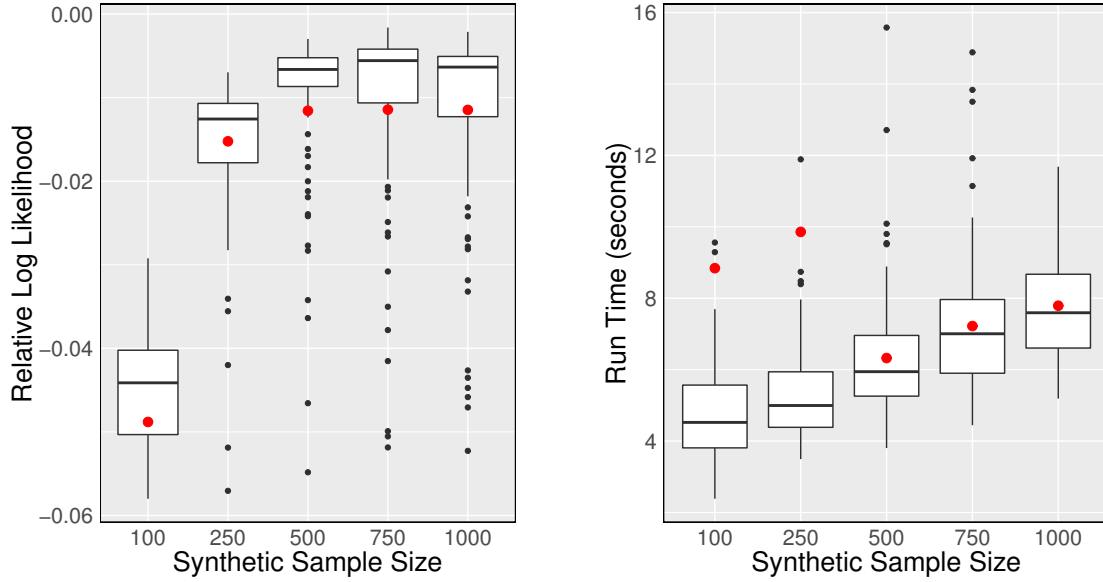


Figure 3.7. The boxplots show the effects of increasing m on the RLL and run time during the Initial Estimate Phase on a univariate sample from f^0 . These boxplots show evidence in favor of larger m values.

The two-dimension results clearly favor a large m , as the boxplots in Figure 3.9 show evidence that the RLL and run times improve as m increases. We expect the run time to increase as m increase since the number of constraints increases as m increases. The decrease in run time points to the problem becoming better conditioned, which requires the nonlinear optimization solver fewer iterations to reach a solution.

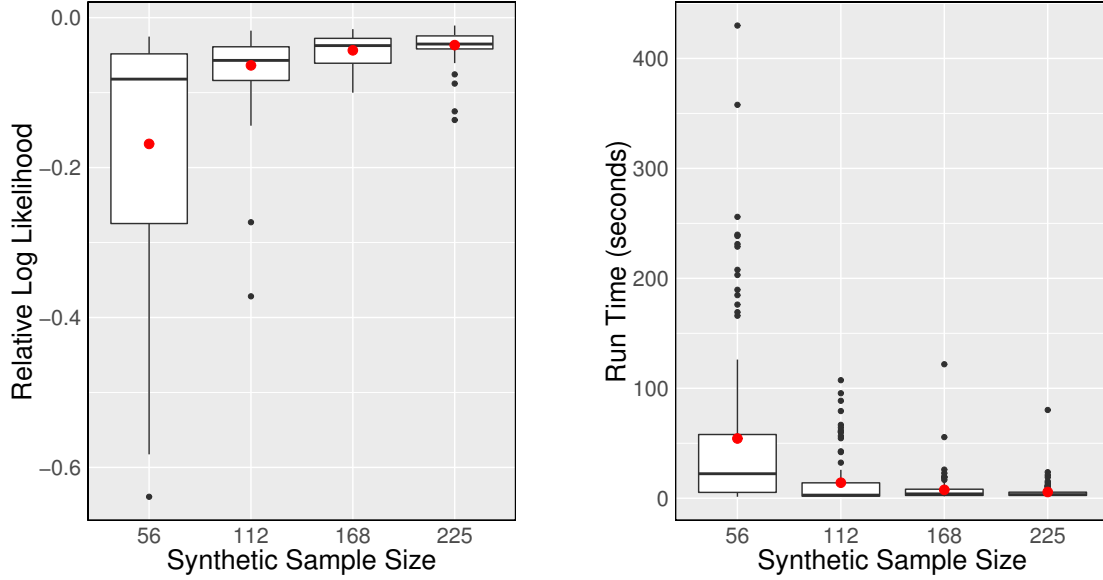


Figure 3.8. The boxplots show the effects of increasing m on the RLL and run time during the Initial Estimate Phase on a bivariate sample from f^0 . Similar to the univariate experiment, these boxplots also show evidence in favor of larger m values.

We adopt the convention of generating the same number of ξ^j points as the sample size n . Further numerical analysis is warranted to explore if setting $m > n$ can be advantageous, and if the results in one and two dimensions hold for higher dimensions.

3.4 Combined Effects

We keep γ , λ , and the number of components, q and r , constant while testing the initial estimate methodology, parallel processing, and the synthetic data size, m , because we assume they each have independent effects based on the intuition gained developing the algorithm described in Chapter 2; further analysis is required to validate this assumption. We now examine the combined effects of γ , λ , q and r during the Initial Estimate Phase in one, two and three dimensions. Recall from Chapter Two, the SMLP Algorithm adapts the γ value to prevent overflow errors, and to minimize the approximation error (2.6). Even though γ changes over the course of the SMLP Algorithm, the initial γ can affect the run time and solution quality. The choice of λ moderates the regularization, π , on the objective

function. A large λ can render the problem infeasible by making the estimate too inflexible, however an insufficiently small λ can allow the estimate to become overly sensitive to the sample. Increasing q and r may improve the estimate quality; however, it can prolong run times.

We construct a 17 design point Orthogonal Latin Hypercube (OLH) to examine the effects of various ranges of these parameters; see (Cioppa and Lucas 2007) for a discussion on testing less than seven factors with 17 design points. For simplicity, we keep q and r equal, and allow them to range from 5 to 20. We allow γ to range from 50 to 300 and λ to range from 0.1 to 1 for the one- and two-dimension experiment. We modify the design for the three-dimension test to procure timely results. We execute the Initial Estimate Phase on 30 random 100-point samples from f^0 for each design point, for a total of 510 potential observations in each dimension (we omit observations resulting in infeasibility). Based on the preceding results, we implement the "All at Once" method, use parallel processing, and set $m = 100$ for each experiment.

We intend to discover general trends on the combined parameter effects, not necessarily to produce a prediction model that can pinpoint optimal parameter values. Data analysis methods can be implemented in future work when solving the SMLP to estimate the probability density of actual data. Here, we seek to identify general trends in how the parameters affect the SMLP and run time.

3.4.1 Univariate Designed Experiment Results

The chosen parameter ranges achieve RLL values centered on zero, indicating that the estimate log likelihood on the sample is nearly as high as the actual density. Figure 3.9 shows a slight RLL improvement as q and r increase, along with a rapid run time increase.

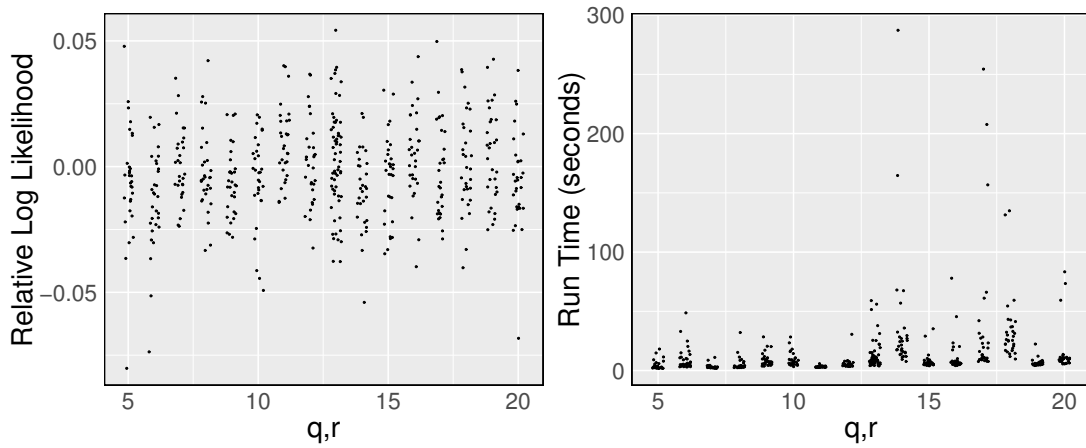


Figure 3.9. During the univariate experiment, the run times increase substantially for $q, r \geq 12$, whereas the RLL generally stabilizes for $q, r \geq 10$. This indicates that allowing q, r to exceed 10 has little potential merit.

Figure 3.10 suggests that γ should be set as low as possible, as there is a slight decline in RLL and a rapid increase in run time as γ exceeds 200. Conversely, λ should take higher values, as the results show slight RLL and run time improvements as λ increases. The reduced run times are expected since higher λ values reduce the estimate flexibility, causing the optimization to converge more rapidly, however the improved RLL is unexpected. A low γ and high λ appear ideal, which together moderate the f_γ slope coefficients, preventing the estimate from incorrectly appropriating high density based on the sample. This moderation may explain why higher λ values improve the RLL.

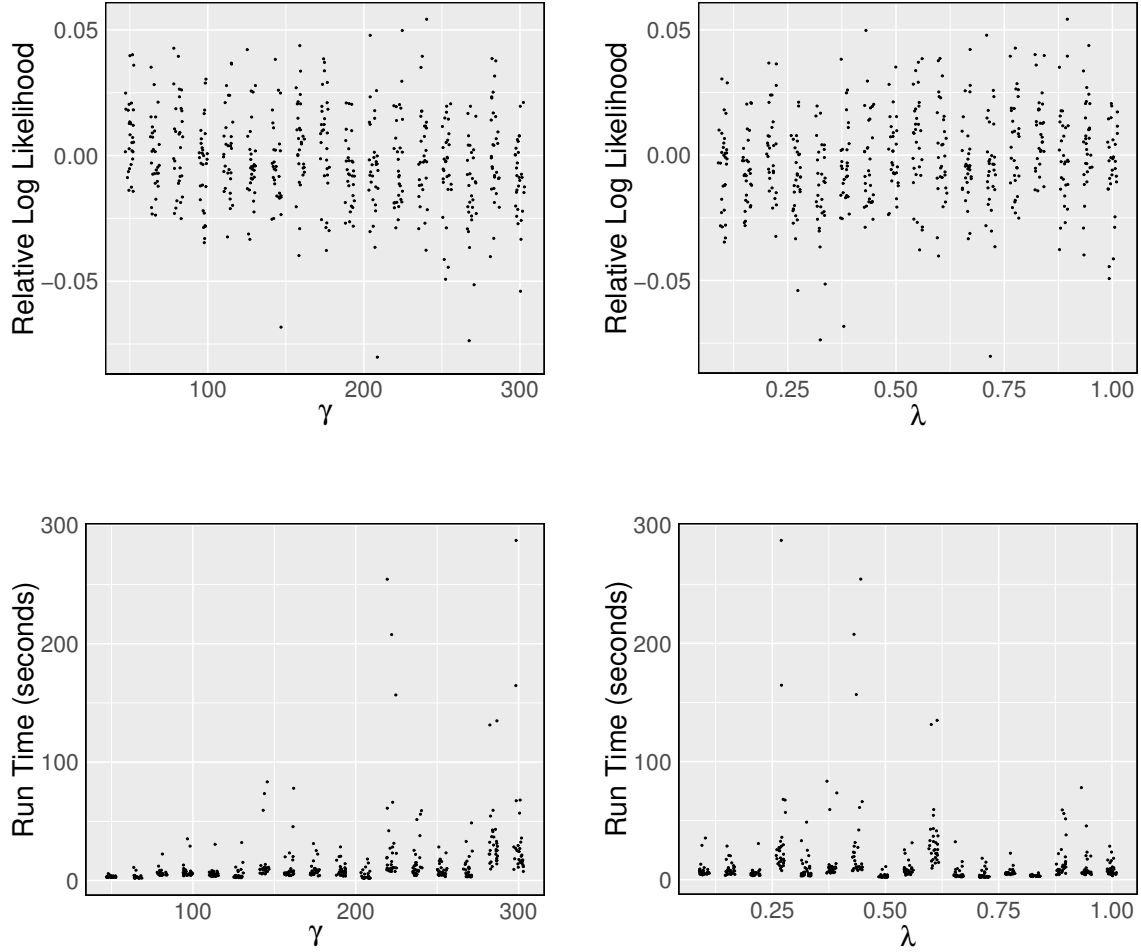


Figure 3.10. This evidence shows that γ and λ have generally opposite effects on the RLL and run time when $d = 1$.

3.4.2 Bivariate Designed Experiment Results

Results of the OLH indicate a slight degradation in density estimate quality as we move from one to two dimensions with the same sample size. Figure 3.10 shows that observations come close to an RLL of zero, however other RLL values are as low as -0.12. This indicates that the estimate quality ranges from having a log likelihood as good as the actual density to a log likelihood that is almost 12% worse.

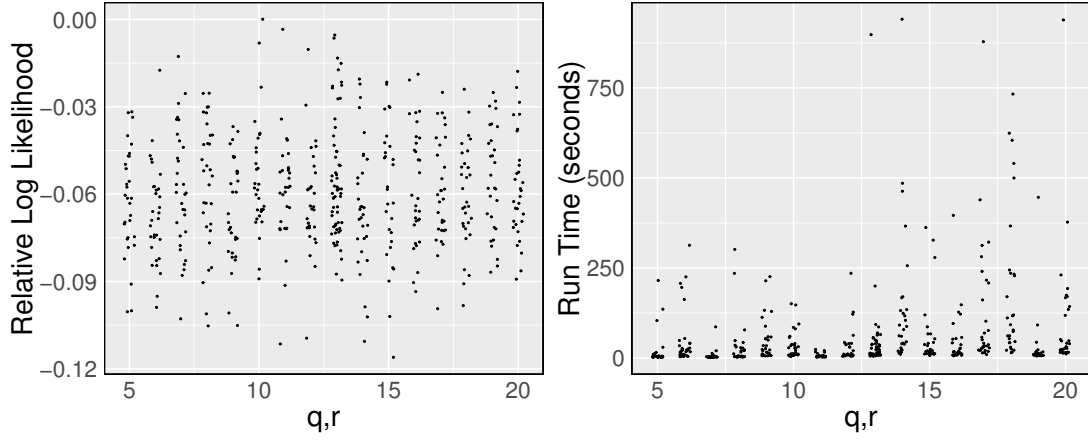


Figure 3.11. The number of piecewise affine functions has little effect on the RLL, however can severely increase the run time for $q, r \geq 12$ when $d = 2$.

Similar to the one-dimension results, Figure 3.12 shows advantages for lower γ values and higher λ values. The results in this experiment are slightly less pronounced than the one-dimension results, as the RLL only ranges from just under zero to -0.09 for all values of γ . There is not strong evidence showing that λ affects the RLL; however, the run time decreases from almost 800 seconds to under two minutes for larger λ values.

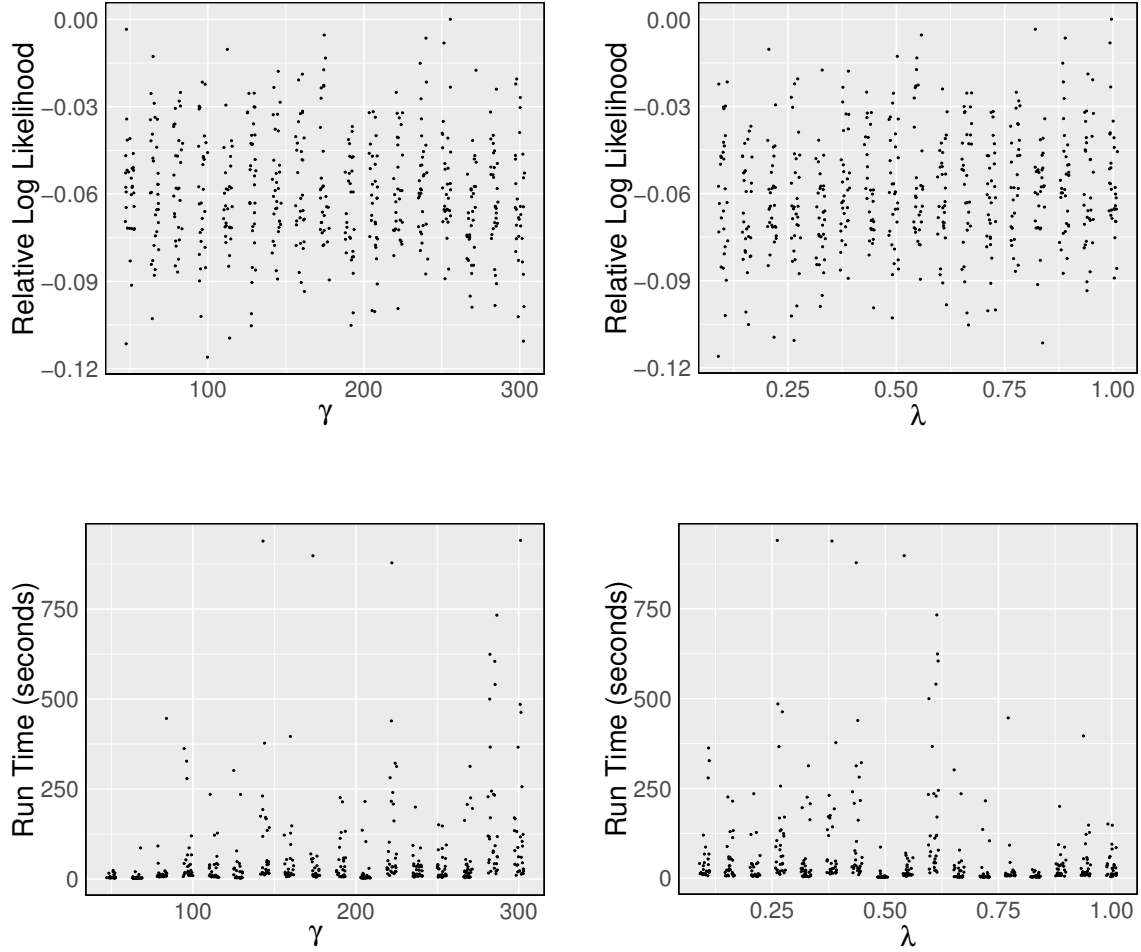


Figure 3.12. Similar to the univariate results, γ and λ have generally opposite effects on the RLL and run time when $d = 2$.

The one-dimension and two-dimension trends resulting from the preceding experiments are very similar. We adopt the same ideal parameter values of q , $r = 10$, $\gamma = 50$, and $\lambda = 1$ for both one-dimension and two-dimension Improvement Search Phase experiments.

3.4.3 Three-Dimension Designed Experiment Results

We modify the experiment design for three dimensions by allowing λ to take values from 10 to 20. Empirical evidence indicates that this is an appropriate range, especially when it

comes to curtailing run times.

Figure 3.13 shows that the RLL decreases to generally under -0.16 during the three dimension experiment, likely due to the well-known challenge in estimating scarce sample densities in multiple dimensions. Similar to both the one- and two-dimension result, Figure 3.14 shows no obvious effects on the RLL for the number of pieces, and indicates a distinct chance of increased run times for $q, r \geq 12$.

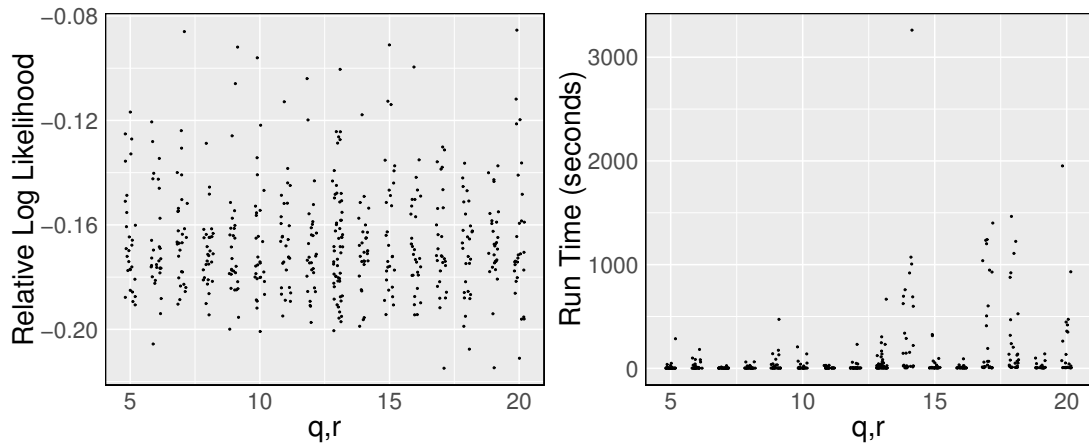


Figure 3.13. In contrast with the univariate and bivariate experiments, the RLL is generally lower and run times are generally higher for the three dimension experiment. The decrease in RLL may be caused by the decrease in coverage of 100 points in three dimensions challenging the estimate ability to attain a high log likelihood. The increase run time is likely caused by the expanding problem size.

The three-dimension λ and γ effects on the RLL are less pronounced than exhibited by the one- and two-dimension results, as shown in Figure 3.14. Based on the results from this experiment, we set $q, r = 10$, $\gamma = 50$, and $\lambda = 20$ for proceeding experiments in three dimensions.

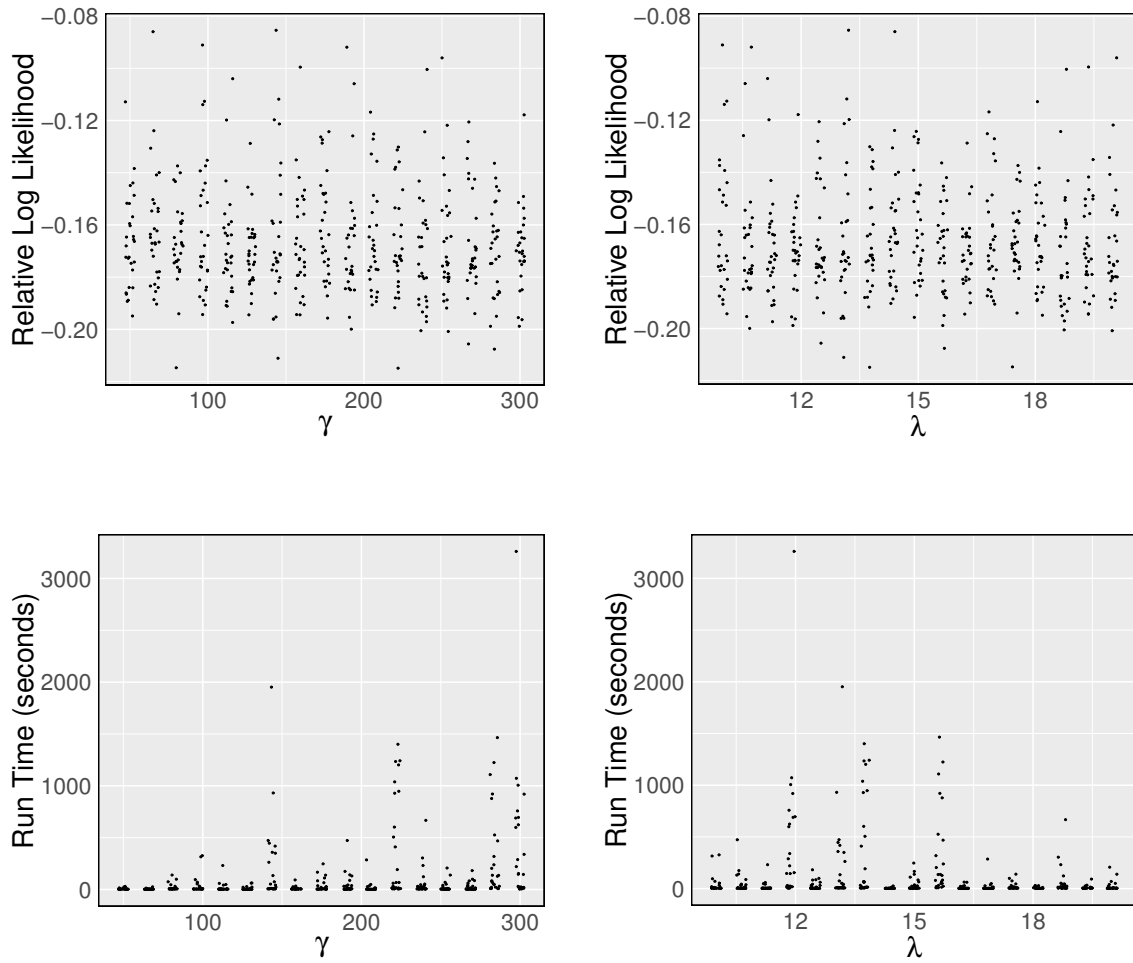


Figure 3.14. When $d = 3$, lower γ values and larger λ values appear slightly more advantageous, mostly due to their effects on the run time.

The overall results from these experiments indicate that run times increase much faster than the model size as the dimension grows. The number of decision variables increases by 20 as the dimension increases from 1 to 3, whereas the run time margin of increase grows from 75 to 375 seconds. This run time expansion can be attributed to an exponential growth in local minima and stationary points.

3.5 Improvement Search Phase Experiments

We now examine how increasing the dimension affects the Improvement Search Phase utility in finding local solutions with higher log likelihood values. In particular, we measure the time that RLL improvements persist, and the RLL improvement size between the initial and the final solution after 1000 improvement search iterations on 100-point samples from f^0 in one, two and three dimensions.

3.5.1 Univariate Results

The RLL increases modestly from 0.011 to 0.014 after 1000 Improvement Search iterations. Figure 3.15 shows diminishing improvements after 30 minutes, with partition three remaining well below the others. This may be due to varying partition sample sizes.

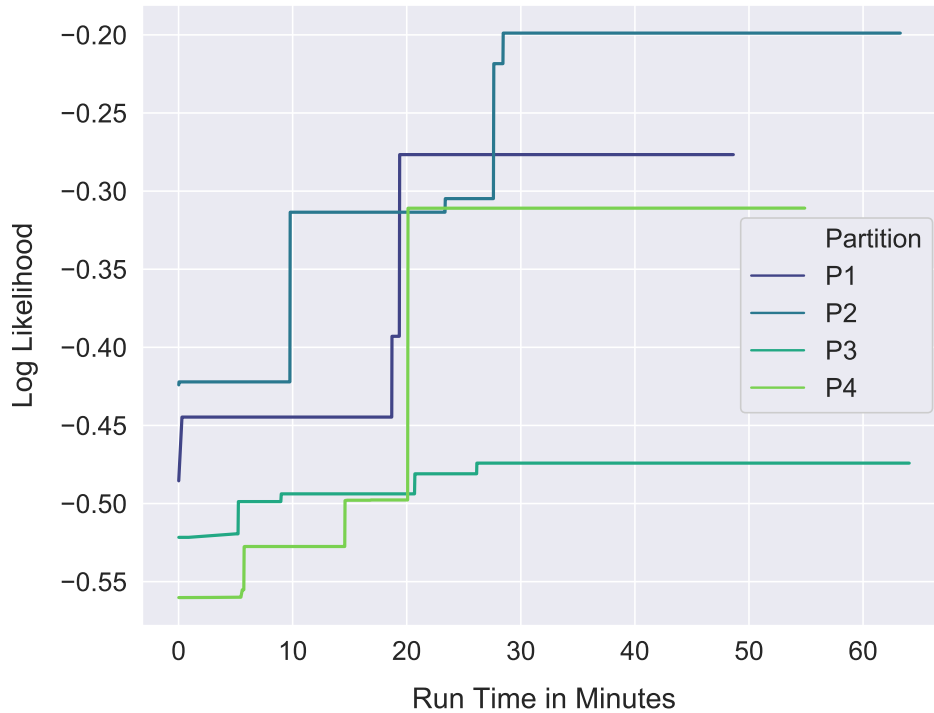


Figure 3.15. The four lineplots track the best log likelihood for each partition estimate. The lineplot lengths indicate the total run time for each estimate to complete 1000 iterations.

Figure 3.16 shows the final estimate graph has peaks and valleys corresponding to segments of high and low data densities. The estimate has a higher log likelihood than the actual density, which is obtained by the sharp peaks in the graph. Increasing the λ regularization parameter may moderate these peaks at the expense of decreasing the RLL.

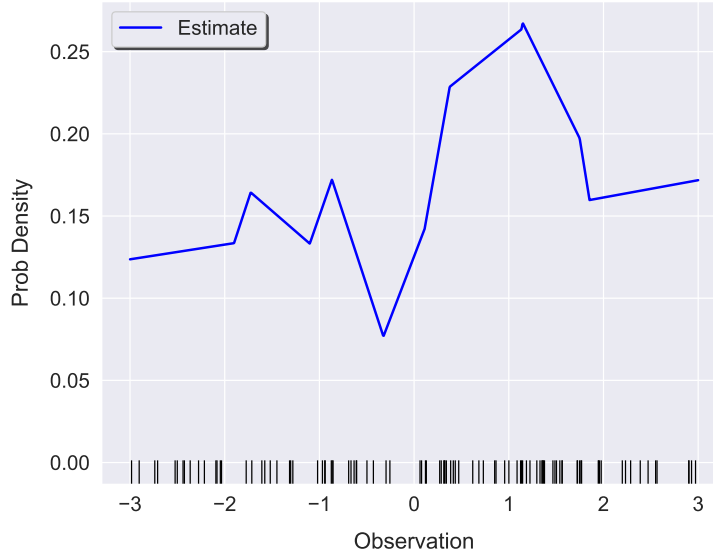


Figure 3.16. This density estimate lineplot is overly sensitive to the sample, indicative of a positive RLL.

The RLL for the Gaussian kernel density with default bandwidth settings from (Jones et al. 2001) is -0.0372, slightly lower than both the initial and final SMLP estimates. This is a crude kernel estimate representation since bandwidth tuning is omitted. However, this demonstrates that the density estimator we are developing yields comparable results to widely used methods.

3.5.2 Bivariate Results

The RLL improves from -0.005 to 0.076 in just under two hours during the bivariate experiment. This indicates that the Improvement Search can still locate improved solutions with the same sample size in two dimensions, however the run time nearly doubles, likely due to the exponential growth in local minima and stationary points. Figure 3.17 shows rapid log likelihood improvements for all partitions within the first 20 minutes of computation.

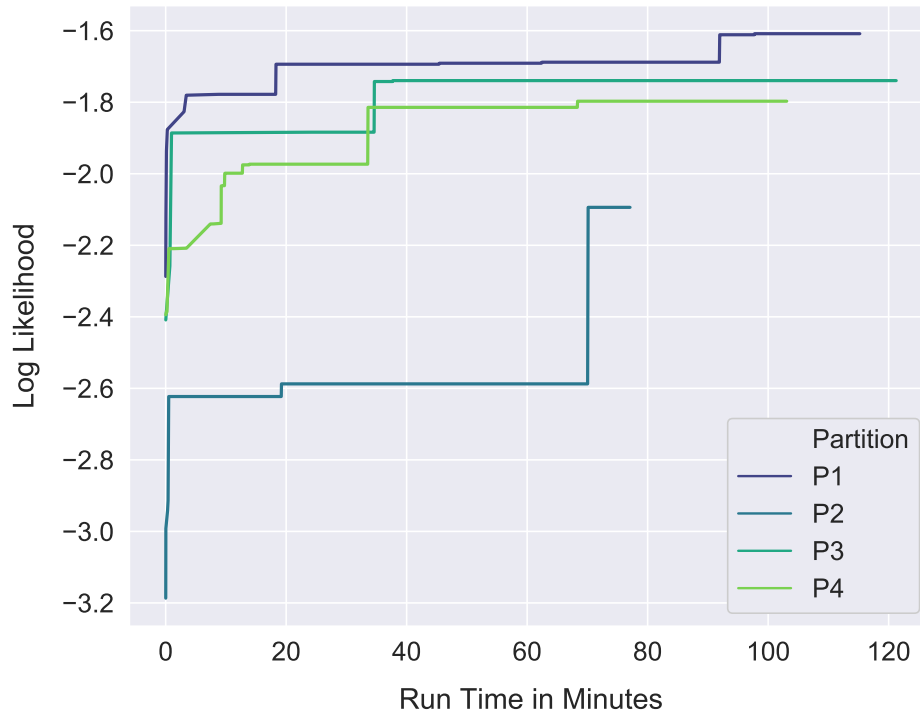
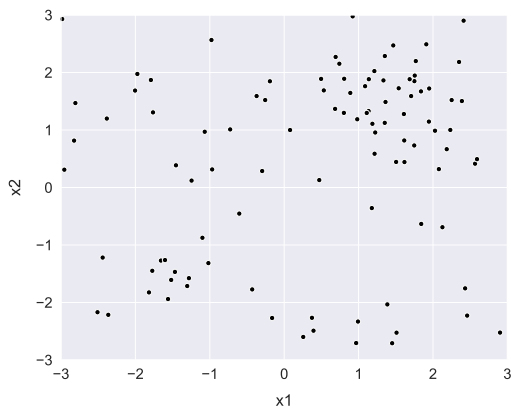
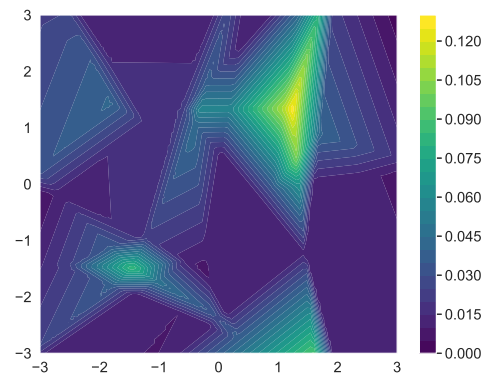


Figure 3.17. Partition two does not improve significantly until after one hour, and never becomes as high as the other partitions. This may be due to a relatively smaller sample size in this partition limiting the maximum attainable log likelihood value.

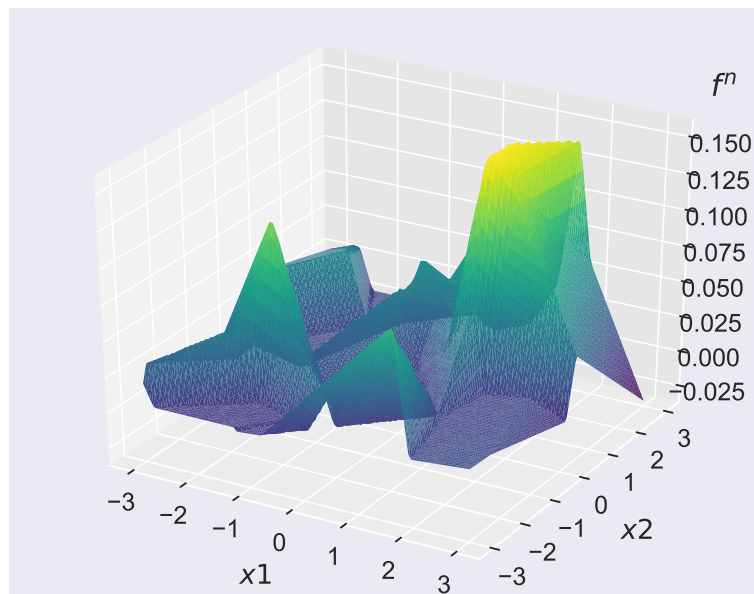
The final estimate surface in Figure 3.18 exhibits peaks and valleys corresponding to high and low densities in the sample scatter plot. In particular, recall the high-density region from Figure 3.1 in the upper right corner. Figure 3.19 shows that the estimate appropriately allocates sufficient density to that location above all others.



Bivariate Sample Data



Density Estimate Contour Plot



Density Estimate Surface Plot

Figure 3.18. The contour plot shows four high density regions corresponding to the four data clusters in the scatter plot. The surface plot shows the two-dimensional piecewise affine function density estimate.

The Gaussian kernel with the default bandwidth from (Jones et al. 2001) RLL is -0.128 , which again shows evidence that the utility of the density estimator we are developing is comparable to widely used methods.

3.5.3 Three-Dimension Results

The three-dimension RLL increases from -0.173 to -0.04 , indicating that the initial estimate is almost 20% worse than the actual density log likelihood, and the final estimate improved to within 4%. The final RLL remains negative after 1000 improvement search iterations, showing the effects of increasing the dimension on the log likelihood. Similar to the two-dimension result, Figure 3.19 shows diminishing improvements for all partitions after 20 minutes of computation.

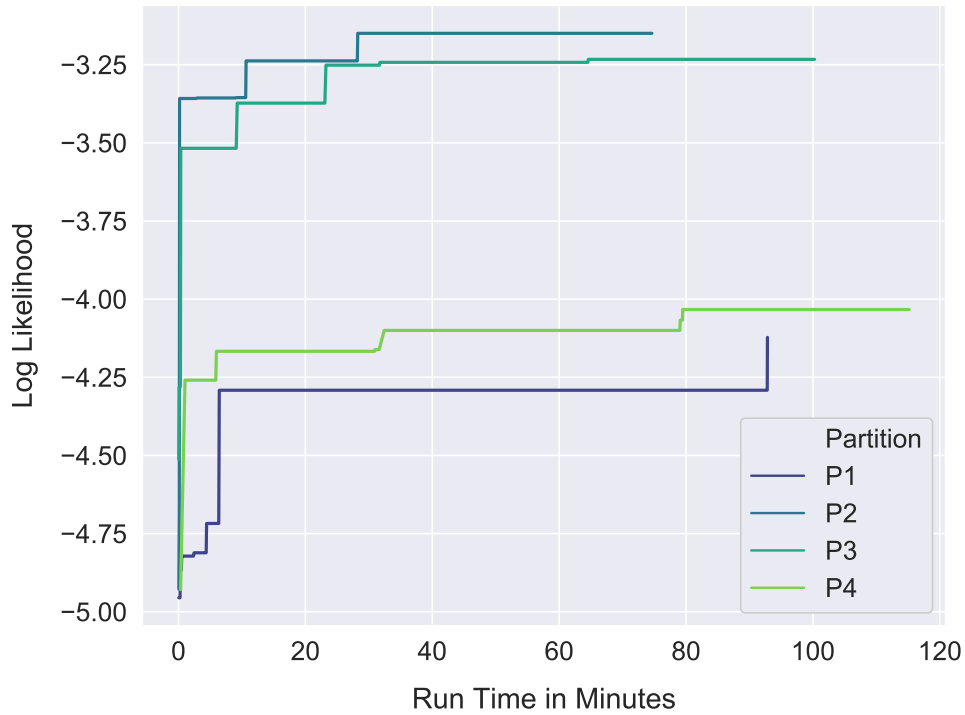


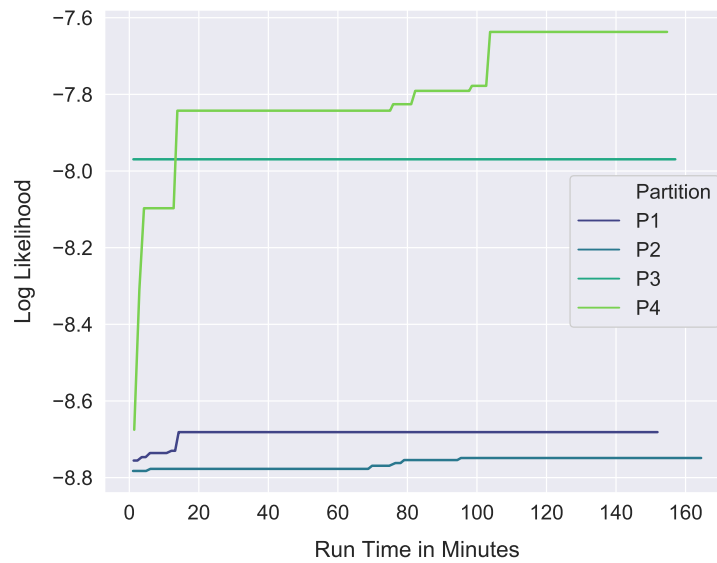
Figure 3.19. We see two partitions with consistently lower log likelihood values throughout the experiment. This may be attributed to the relatively scarce sample size of 100 points in three dimensions challenging these two partitions to attain a high log likelihood values.

The combined results of the three Improvement Search experiments indicate that the RLL improvement diminishes after 20 minutes in each dimension. The results also indicate that although the SMLP does not rely on a mesh, increasing the dimension adversely affects the estimate log likelihood. We also observe the run time increasing with the dimension, which we expect due to the linear increase in problem size.

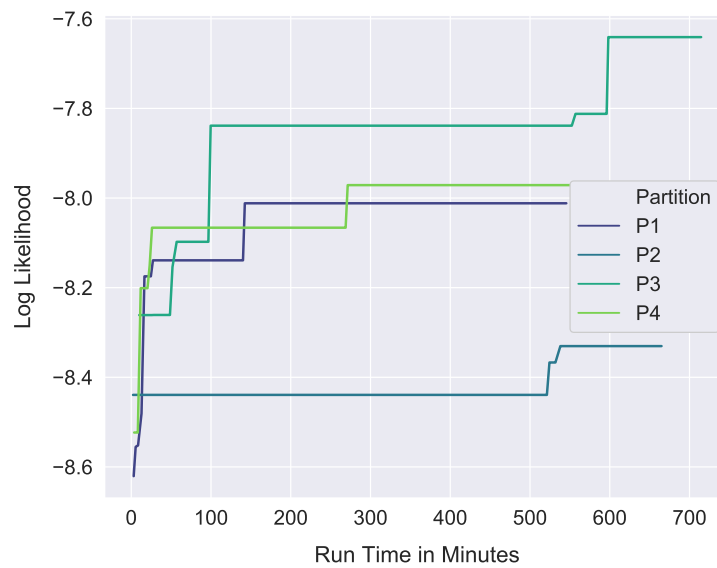
3.6 Five-Dimension Experiments

We now study how varying the sample size in five dimensions affects the Improvement Search Phase run time and RLL. We conduct 100 improvement search iterations on a 100- and 500-point five-dimension sample from f^0 . We expect the 500-point sample to have larger run times due to the problem constraint increase. We are interested in determining if the larger sample can offer increased RLL improvements, along with prolonging the run time.

The RLL increases from -0.293 to -0.224 in the 500 point experiment and only rises from -0.293 to -0.273 in the 100-point experiment, indicating that the larger sample has very little positive effect on the Improvement Search Phase ability to find improved solutions. In striking contrast, the run times considerably increase. Figure 3.20 shows that the 100-point experiment completed 100 improvement search iterations in under three hours, whereas the 500-point experiment took over 12 hours to complete the same number of improvement iterations.



Improvement Search Results for $n = 100$



Improvement Search Results for $n = 500$

Figure 3.20. Increasing the sample size from 100 to 500 has little positive effect on the RLL, however dramatically prolongs the run time.

This evidence indicates that the capability of our density estimate to achieve a high log

likelihood is not greatly affected by scarce samples. The run time however is substantially increased, especially for higher than two dimensions.

CHAPTER 4:

Conclusions

This thesis develops and evaluates a new multivariate density estimator that seeks to improve the flexibility to represent nuances in sample data, and to prevent the problem from growing exponentially in d . Our estimation method accomplishes both. We expand the class of candidate functions to include virtually any probability density function, and our formulation grows linearly in d . However, our method poses a nonconvex optimization problem, which may become ill-conditioned. In addition, our method relies on nine parameters, which our investigation shows can affect the estimate quality and run time. Despite these challenges, this work demonstrates the viability of using piecewise affine functions to estimate multivariate distributions up to five dimensions, on sample sizes as low as 100 points.

The most substantial disadvantage to our method is the run time that increases exponentially as d increases and as the number of components increase. Our results show that increasing the sample size only modestly improves the ability for the estimate to obtain improved log likelihood values. They also show that as the sample increases, especially when $d > 3$, the nonlinear optimization solver requires exponentially more iterations to reach a solution.

Further research is required to develop this nascent estimation method. The use of a perturbation method during the Improvement Search Phase requires additional testing, as do alternate strategies for locating improved solutions faster. Also, testing the effects of setting $m > n$ and sampling more ξ values than the data sample size may prove beneficial. The results show improvement in RLL and run time as m increases; however, this improvement diminishes as m approaches n . Increasing m may enhance the estimator ability to find a solution, especially for scarce sample sizes, as the problem becomes better conditioned with an improved integral estimate. Other parameters require additional research as well to further investigate their combined effects.

Despite run time challenges and pending developmental research requirements, our results show that this new density estimation method offers potential advantages over current methods, and has the potential to scale to higher dimensions.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: Computation

We perform all computations on a 2015 Macintosh laptop with a 2.7 GHz Intel Core i5 processor and 8 GB RAM using the Python computer language. We implement all nonlinear programs in the Pyomo (Hart et al. 2017) environment using the Interior Point Optimizer software (Lougee-Heimer 2003) solver. We use the following solver optimality conditions.

Maximum iterations: 700000000

Tolerance: 0.001

Acceptable Tolerance: 0.005

Acceptable Iterations: 15

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

- Birke M (2009) Shape constrained kernel density estimation. *Journal of Statistical Planning and Inference* 139:2851–2862.
- Brown G, Dell R (2007) Formulating linear programs: A rogues’ gallery. *INFORMS Trans. on Educa.* 7(2):153–159, URL <http://dx.doi.org/10.1287/ited.7.2.153>.
- Chang M, Sangrey P (2019) Bypassing the curse of dimensionality: Feasible multivariate density estimation URL http://sangrey.io/feasible_multivariate_density_estimation.pdf.
- Cioppa T, Lucas T (2007) Efficient nearly orthogonal and space-filling latin hypercubes. *Technometrics* 49:45–55, URL <http://dx.doi.org/10.1198/004017006000000453>.
- Cule M, Gramacy R, Samworth R (2009) LogConcDEAD: An R package for maximum likelihood estimation of a multivariate log-concave density. *Journal of Statistical Software* 29(2), URL <http://www.jstatsoft.org/v29/i02/>.
- Cule M, Samworth R (2010) Theoretical properties of the log-concave maximum likelihood estimator of a multidimensional density. *Electron. J. Statist.* 4:254–270, URL <http://dx.doi.org/10.1214/09-EJS505>.
- Fix E, Hodges JL Jr (1951) Discriminatory analysis - nonparametric discrimination: Consistency properties. Technical report, URL <http://handle.dtic.mil/100.2/ADA800276>.
- Hart W, Laird C, Watson J, Woodruff D, Hackebeil G, Nicholson B, Siirola J (2017) *Pyomo—optimization modeling in python* (Springer Science & Business Media).
- Jones E, Oliphant T, Peterson P, et al. (2001) SciPy: Open source scientific tools for Python. URL <http://www.scipy.org/>, [Online; Accessed May 27, 2019].
- Liu L, Wong WH (2014) Multivariate density estimation via adaptive partitioning (i): Sieve mle <https://arxiv.org/abs/1401.2597>.
- Lougee-Heimer R (2003) The common optimization interface for operations research. *IBM Journal of Research and Development* 47:57–66.
- Nagler T, Czado C (2016) Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas. *Journal of Multivariate Analysis* 151(2):69–89.
- Polak E, Royset JO, Womersley R (2003) Algorithms with adaptive smoothing for finite minimax problems. *Journal of Optimization Theory and Applications* 119(3):459–484.

- Royset JO (2017) Approximations of semicontinuous functions with applications to stochastic optimization and statistical estimation. *arXiv:1709.06730* .
- Royset JO (2019) Nonlinear Program class lecture at Naval Postgraduate School, 2019-05-15.
- Royset JO, Wets RJB (2015) Fusion of hard and soft information in nonparametric density estimation. *European Journal of Operational Research* 247(2):532–547.
- Royset JO, Wets RJB (2016) Multivariate epi-splines and evolving function identification problems. *Set-Valued and Variational Analysis* 24(4):517–545, erratum, pp. 547-549.
- Royset JO, Wets RJB (2017) Variational Analysis of Constrained M-Estimators, in review ArXiv:1702.08109.
- Sankaran P, Unnikrishnan Nair N (2009) Nonparametric estimation of hazard quantile function. *Journal of Nonparametric Statistics* 21(6):757–767.
- Seregin A, Wellner JA (2010) Nonparametric estimation of multivariate convex-transformed densities. *Annals of statistics* 38(6):3751.
- Silverman BW (1982) On the estimation of a probability density function by the maximum penalized likelihood method. *The Annals of Statistics* 795–810.
- Teter MD, Royset JO, Newman AM (2016) Modeling uncertainty of expert elicitation for use in risk-based optimization. *Annals of Operations Research* 1–22.
- Thompson E (2013) Six blind men the organizational elephant: Microscopic and telescopic versions of the fundraising narrative. URL <https://ceplan.com/the-organizational-elephant>.

Initial Distribution List

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California