

Modeling and Estimation Techniques for Wide-Area Network Traffic with Atypical Components

Carl E. Minton

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Scott F. Midkiff, Chair
Nathaniel J. Davis, IV
Ira Jacobs

April 25, 2002
Blacksburg, Virginia

Keywords:
atypical network traffic, contamination detection, multimodal estimation

Copyright © 2002, Carl E. Minton

Modeling and Estimation Techniques for Wide-Area Networks with Atypical Components

Carl E. Minton

(ABSTRACT)

A critical first step to improving existing and designing future wide-area networks is an understanding of the load placed on these networks. Efforts to model traffic are often confounded by atypical traffic — traffic particular to the observation site not ubiquitously applicable. The causes and characteristics of atypical traffic are explored in this thesis. Atypical traffic is found to interfere with parsimonious analytic traffic models. A detection and modeling technique is presented and studied for atypical traffic characterized by strongly clustered inliers. This technique is found to be effective using both real-world observations and simulated data.

Another form of atypical traffic is shown to result in multimodal distributions of connection statistics. Putative methods for bimodal estimation are reviewed and a novel technique, the midpoint-distance profile, is presented. The performance of these estimation techniques is studied via simulation and the methods are examined in the context of atypical network traffic. The advantages and disadvantages of each method are reported.

Acknowledgements

This thesis would not be, but for the contributions of numerous individuals. I would like to express my gratitude to all of them, fully accepting the risk of babbling like an award recipient at the Academy Awards.

Foremost, my advisor, Dr. Scott Midkiff, has proved an invaluable mentor. This work has his insight, patience, and occasional prodding to thank. His dedication to the academic profession has not gone unnoticed, and it is truly appreciated. I would also like to thank the other members of my committee, Dr. Nat Davis and Dr. Ira Jacobs, for their support and suggestions.

Chapter 5 of this work grew out of the exposure to robust estimation afforded to me by Dr. Lamine Mili. His interest in the subject proved contagious, and for the exposure to the field I am grateful.

The research presented in this work would not have been possible without the contributions of the Via family to the Bradley Department of Electrical and Computer Engineering. Their philanthropy provided me with the latitude to pursue research free from worries on the basal end of Maslow's hierarchy.

During my research, and time following, I have been fortunate to have the support and cajoling of my friends. They have been valued distractions from and sounding boards for the material herein. Thanks to Jason, Jeff, Pete, Tom, Dave, (the other) Jeff, (the other) Dave, Paul, Scott, and (still one more) Jeff.

The warmest gratitude possible is expressed to my wife, Amy, whose support during the final drafting of this thesis was remarkable. Her patience while I ignored her to finish writing will be remembered when she begins writing her dissertation.

Finally, this work is dedicated to the memory of my mother, Sarah Jeanne Carlson Minton. Her good humor, resolve, and kindness in spite of tremendous adversity are an example I will endeavor to follow throughout my life. Although she did not live to see the completion of this thesis, I'm sure her reaction would have been mixed: mostly pride with a little impatience for "taking long enough."

Table of Contents

LIST OF VARIABLES AND NOTATIONS	VII
LIST OF FIGURES.....	IX
LIST OF TABLES.....	XI
CHAPTER 1: INTRODUCTION	1
1.1. ORGANIZATION	1
CHAPTER 2: BACKGROUND.....	3
2.1. NETWORK TRAFFIC MODELING TECHNIQUES	3
2.1.1. <i>Empirical Models of WAN Traffic</i>	3
2.1.2. <i>Empirically Derived Analytic Models</i>	4
2.1.3. <i>Self-Similar Traffic Models</i>	5
2.2. BIMODAL NETWORK TRAFFIC.....	5
2.3. PARAMETER ESTIMATION	7
2.3.1. <i>Estimator Properties</i>	7
2.3.2. <i>Maximum Likelihood Estimation</i>	8
2.3.3. <i>Method of Moments</i>	9
2.4. MEASURING QUALITY OF FIT.....	10
2.5. ROBUST ESTIMATION	12
2.5.1. <i>Midpoint of the Shortest Half</i>	13
2.5.2. <i>Length of the Shorth</i>	14
2.6. SUMMARY.....	15
CHAPTER 3: THE NATURE OF ATYPICAL TRAFFIC	16
3.1. DEFINITION OF ATYPICAL TRAFFIC	16
3.2. SOURCES OF ATYPICAL TRAFFIC.....	16
3.2.1. <i>Abnormal User-Level Applications</i>	17
3.2.2. <i>Intrinsic Protocol Errors</i>	17
3.2.3. <i>Site Specific Network Configuration</i>	18
3.3. EFFECT OF ATYPICAL TRAFFIC.....	19
3.3.1. <i>Qualitative Effect of Atypical Traffic</i>	19
3.3.2. <i>Quantitative Effect of Contamination on Model Tractability</i>	22
3.4. CONCLUSIONS	22
CHAPTER 4: DETECTION ALGORITHM FOR ATYPICAL WAN TRAFFIC	24
4.1. ALGORITHM DEFINITION AND OBJECTIVES.....	24
4.2. NOTATION AND TERMINOLOGY.....	25
4.3. ALGORITHM DESCRIPTION	26
4.3.1. <i>Censoring Data Based on Contamination Models</i>	28
4.3.2. <i>Single-Variate Contamination Region Detection</i>	28
4.3.2.1. <i>Contamination Model Formalization</i>	31
4.3.3. <i>Contaminate Model Merging</i>	32
4.3.4. <i>Incorporation of C^H into C</i>	34

4.4.	ALGORITHM IMPLEMENTATION.....	34
4.5.	EVALUATION AND PERFORMANCE.....	35
4.5.1.	<i>Correctness</i>	35
4.5.2.	<i>Stability</i>	38
4.5.3.	<i>Model Minimality</i>	39
4.5.4.	<i>Effect of Parameters on Contamination Detection</i>	39
4.5.4.1.	<i>Meaning of Parameters</i>	39
4.5.4.2.	<i>Effect of Critical Angle on Detection</i>	41
4.5.5.	<i>Scalability</i>	47
4.6.	CONCLUSIONS.....	50
CHAPTER 5:	ESTIMATION OF MULTIMODAL DISTRIBUTIONS	52
5.1.	ATYPICAL TRAFFIC AND MULTIMODAL ESTIMATION	52
5.2.	CONVENTIONAL ESTIMATION TECHNIQUES	53
5.2.1.	<i>Two-Component Gaussian Mixture</i>	54
5.2.2.	<i>Method of Moments</i>	54
5.2.3.	<i>Maximum Likelihood Estimation</i>	57
5.3.	MIDPOINT-DISTANCE PROFILE ESTIMATION	58
5.3.1.	<i>Review of Location and Scale Estimation</i>	58
5.3.2.	<i>Definition of the Midpoint-Distance Profile</i>	60
5.3.3.	<i>Estimation of Location</i>	63
5.3.4.	<i>Estimation of Scale</i>	64
5.3.4.1.	<i>Length of the Shorth</i>	64
5.3.4.2.	<i>Scale Estimation using Relative Distances</i>	67
5.3.5.	<i>Estimating Component Weight</i>	71
5.3.6.	<i>Summary</i>	72
5.4.	MIDPOINT-DISTANCE PROFILE IMPLEMENTATION.....	72
5.4.1.	<i>Profile Noise</i>	73
5.4.2.	<i>Location of Regions Near Minima</i>	74
5.4.3.	<i>Estimation of Parameters</i>	75
5.4.4.	<i>Summary</i>	77
5.5.	ESTIMATOR PERFORMANCE	77
5.5.1.	<i>Bias</i>	77
5.5.2.	<i>Rate of Convergence</i>	78
5.5.3.	<i>Efficiency</i>	79
5.5.4.	<i>Numerical Complexity</i>	79
5.5.5.	<i>Parametric Sensitivity and Effects</i>	80
5.5.6.	<i>Extensibility</i>	81
5.6.	APPLICATION TO WAN MODELING.....	82
5.7.	SUMMARY.....	84
CHAPTER 6:	CONCLUSION.....	86
6.1.	SUMMARY.....	86
6.2.	FUTURE WORK.....	87
REFERENCES		89
APPENDIX A:	WAN CONNECTION MEASUREMENT EDFs.....	93

A.1.	TELNET TRAFFIC	93
A.2.	NNTP TRAFFIC — EFFECT OF ATYPICAL TRAFFIC	95
A.3.	NNTP TRAFFIC — EFFECT OF CO-LOCATION.....	97
A.4.	SMTP TRAFFIC.....	99
APPENDIX B:	MODEL MERGING TECHNIQUE.....	101
B.1.	EXAMPLE OF MODEL MERGING TECHNIQUE.....	101
B.2.	INSTABILITY OF MERGING ON ALL MODELS	102
B.3.	COMPARISON OF MERGING BASED ON ALL VERSUS WORKING DATA..	105
APPENDIX C:	MATLAB SOURCE FOR CONTAMINATION DETECTION	
ALGORITHM	112	
C.1.	OVERALL ALGORITHM	112
C.2.	DATA CENSORING	113
C.3.	SINGLE-VARIATE CONTAMINATION DETECTION.....	113
C.4.	MODEL MERGING.....	116
APPENDIX D:	SCALE INVARIANCE OF DISTANCE RATIO.....	117
APPENDIX E:	MATLAB SOURCE FOR MULTIMODAL ESTIMATION	
TECHNIQUES	118	
E.1.	GAUSSIAN MIXTURE GENERATION.....	118
E.2.	METHOD OF MOMENTS.....	119
E.3.	MAXIMUM LIKELIHOOD ESTIMATION.....	120
E.4.	MIDPOINT DISTANCE PROFILE	121
APPENDIX F:	MULTIMODAL ESTIMATION SIMULATION RESULTS...	124
F.1.	SIMULATION EXPERIMENT	124
F.2.	ESTIMATOR BIAS.....	124
F.3.	ESTIMATOR RATE OF CONVERGENCE	131
F.4.	ESTIMATOR RELATIVE EFFICIENCY	134
VITA		136

List of Variables and Notations

General Notations

Z	Random variable
$f_z(z)$	Probability density of random variable Z at value z
$F_z(z)$	Cumulative distribution function of random variable Z at value z
$E[Z]$	Expected value of random variable Z
$\text{var}[Z]$	Variance of random variable Z
$P[G]$	Probability of random event G
$\tilde{Z}$	Observation set of random variable Z
$\Theta(\tilde{Z})$ or m	Size of the observation set
$z_{[i]}$	The i^{th} order statistic of the observation set $\tilde{Z}$
θ	A parameter of a distribution function of a random variable
$\bar{\theta}$	The set of parameters of a distribution function
$\hat{\theta}_{MLE}$	Maximum likelihood estimates of a parameter set
$\hat{\theta}(\tilde{Z})$ or $\hat{\theta}$	An estimator of the observation set
I_f	Fisher information
M_n	The n^{th} analytic moment of a distribution function
$\hat{M}_n$	The n^{th} estimated moment of an observation set
M'_n	The n^{th} analytic central moment of a distribution function
$\hat{M}'_n$	The n^{th} estimated central moment of an observation set
$\hat{\lambda}^2$	The Pederson-Johnson model discrepancy estimate
$\Theta(S)$	Number of elements of set S
$\phi(z)$	Standard Gaussian probability density function at z
$\Phi(z)$	Standard Gaussian cumulative distribution at z
$\Phi^{-1}(p)$	Inverse distribution function of the standard Gaussian variate at p

Contamination Detection Algorithm

D	Dataset of observations
D_w	Working dataset
MD	Metadata of dataset, D
C	Set of contamination models
C_H	Set of hypothesized contamination models
α	Minimum detectable level of contamination
θ_c	Critical angle of contaminate detection
$\psi_{i,j}$	Coverage of model C_i by model C_j
CM	Coverage matrix
ACM	Augmented coverage matrix
γ	Minimum coverage for model merging

Midpoint-Distance Profile Estimation

α	Fraction of the data used in the data window
d_j	The j^{th} distance element of the midpoint-distance profile
mp_j	The j^{th} midpoint element of the midpoint-distance profile
$d_{\min}$	The minimum distance of the midpoint-distance profile
$mp_{\min}$	The midpoint element corresponding to $d_{\min}$
$\mathfrak{R}_i$	The region of the midpoint-distance profile in the neighborhood of the i^{th} minimum
μ_i	Location parameter of the i^{th} component of a multimodal distribution
σ_i	Scale parameter of the i^{th} component of a multimodal distribution
w_i	Weight parameter of the i^{th} component of a multimodal distribution
d_μ	Distance of the data fraction with a midpoint of μ
$\hat{p}(mp)$	Polynomial approximation to the midpoint-distance profile
$\hat{p}_i(mp)$	Binomial approximation to the midpoint-distance profile in $\mathfrak{R}_i$
$\delta(\alpha)$	Linear approximation to the distance ratio

List of Figures

FIGURE 3-1 EDF OF TELNET TRANSMITTED BYTES	20
FIGURE 4-1 DETECTION ALGORITHM OVERVIEW.....	27
FIGURE 4-2 SINGLE-VARIATE DETECTION ILLUSTRATION	31
FIGURE 4-3 ERROR SURFACE, MEAN RATIO = 2.....	43
FIGURE 4-4 OPTIMAL CRITICAL ANGLE, MEAN RATIO=2	44
FIGURE 4-5 ERROR SURFACE, MEAN RATIO = 4.....	45
FIGURE 4-6 AVERAGE ERROR RATE IN NEAR-OPTIMAL REGION	46
FIGURE 4-7 EFFECT OF NUMBER OF VARIABLES ON SCALABILITY	48
FIGURE 4-8 EFFECT OF DATASET SIZE ON SCALABILITY.....	48
FIGURE 4-9 EFFECT OF NUMBER OF MODELS ON FLOPS.....	49
FIGURE 4-10 EFFECT OF NUMBER OF MODELS ON ITERATIONS	50
FIGURE 5-1 NNTP DURATION PDF	53
FIGURE 5-2 COMPARISON OF TWO HALVES.....	59
FIGURE 5-3 COMPARISON OF TWO MIDPOINT-DISTANCE PROFILES.....	62
FIGURE 5-4 MIDPOINT DISTANCE PROFILE FOR A BIMODAL DISTRIBUTION	63
FIGURE 5-5 MINIMUM DISTANCES DEPENDENCY ON COMPONENT WEIGHT	65
FIGURE 5-6 SENSITIVITY OF CORRECTION FACTOR TO COMPONENT WEIGHT	66
FIGURE 5-7 RELATIVE DISTANCES.....	67
FIGURE 5-8 EFFECT OF COMPONENT WEIGHT ON DISTANCE RATIO.....	69
FIGURE 5-9 AVERAGE DISTANCE RATIO AND LINEAR APPROXIMATION	71
FIGURE 5-10 PROBLEM OF NOISE IN A PROFILE	73
FIGURE 5-11 POLYNOMIAL APPROXIMATION TO THE PROFILE	74
FIGURE 5-12 REGIONS OF INFLUENCE FOR A PROFILE.....	75
FIGURE 5-13 BINOMIAL APPROXIMATIONS IN REGIONS OF INFLUENCE.....	76
FIGURE 5-14 NUMERICAL COMPLEXITY	80
FIGURE 5-15 PROFILE OF A THREE COMPONENT MIXTURE	82
FIGURE 5-16 GAUSSIAN MIXTURE FITS TO NNTP DATA.....	83
FIGURE 5-17 GAUSSIAN COMPONENT AS A MODEL FOR REMOTE NNTP CONNECTIONS	84
FIGURE A-1 EFFECT OF ATYPICAL TRAFFIC ON TELNET DURATION	93

FIGURE A-2 EFFECT OF ATYPICAL TRAFFIC ON TELNET RECEIVED BYTES	94
FIGURE A-3 EFFECT OF ATYPICAL TRAFFIC ON TELNET TRANSMITTED BYTES	94
FIGURE A-4 EFFECT OF ATYPICAL TRAFFIC ON NNTP DURATION.....	95
FIGURE A-5 EFFECT OF ATYPICAL TRAFFIC ON NNTP RECEIVED BYTES	96
FIGURE A-6 EFFECT OF ATYPICAL TRAFFIC ON NNTP TRANSMITTED BYTES.....	96
FIGURE A-7 EFFECT OF CO-LOCATION ON NNTP DURATION.....	97
FIGURE A-8 EFFECT OF CO-LOCATION ON NNTP RECEIVED BYTES	98
FIGURE A-9 EFFECT OF CO-LOCATION ON NNTP TRANSMITTED BYTES	98
FIGURE A-10 EFFECT OF ATYPICAL TRAFFIC ON SMTP DURATION.....	99
FIGURE A-11 EFFECT OF ATYPICAL TRAFFIC ON SMTP RECEIVED BYTES	100
FIGURE A-12 EFFECT OF ATYPICAL TRAFFIC ON SMTP TRANSMITTED BYTES.....	100
FIGURE B-1 DATA AFTER DETECTION, MERGING BASED ON ALL DATA.....	110
FIGURE B-2 DATA AFTER DETECTION, MERGING BASED ON WORKING DATA.....	111
FIGURE F-1 AVERAGE ACTUAL BIAS (MLE).....	125
FIGURE F-2 AVERAGE OBSERVATION BIAS (MLE).....	126
FIGURE F-3 AVERAGE ACTUAL BIAS (MOM)	126
FIGURE F-4 AVERAGE OBSERVATION BIAS (MOM)	127
FIGURE F-5 AVERAGE ACTUAL BIAS (MPDP)	127
FIGURE F-6 AVERAGE OBSERVATION BIAS (MPDP)	128
FIGURE F-7 MLE VARIANCE VERSUS DATA SIZE.....	132
FIGURE F-8 MOM VARIANCE VERSUS DATA SIZE	132
FIGURE F-9 MPDP VARIANCE VERSUS DATA SIZE.....	133
FIGURE F-10 RELATIVE EFFICIENCY OF MOM ESTIMATION	134
FIGURE F-11 RELATIVE EFFICIENCY OF MPDP ESTIMATION	135

List of Tables

TABLE 3-1: MEASURING MODELING TRACTABILITY VIA MODEL DISCREPANCY	22
TABLE 4-1 METADATA ELEMENTS.....	26
TABLE 4-2 SIMULATION EXPERIMENT PARAMETERS	42
TABLE 5-1 ESTIMATES FROM TWO PROFILES.....	62
TABLE 5-2 AVERAGE ESTIMATOR BIAS.....	77
TABLE 5-3 ESTIMATOR RATES OF CONVERGENCE	78
TABLE 5-4 ESTIMATOR EFFICIENCY RELATIVE TO MAXIMUM LIKELIHOOD.....	79
TABLE F-1 ACTUAL MLE BIAS	128
TABLE F-2 OBSERVATION MLE BIAS	129
TABLE F-3 ACTUAL MOM BIAS.....	129
TABLE F-4 OBSERVATIONAL MOM BIAS	130
TABLE F-5 ACTUAL MPDP BIAS.....	130
TABLE F-6 OBSERVATIONAL MPDP BIAS.....	131
TABLE F-7 ESTIMATED RATES OF CONVERGENCE	133
TABLE F-8 ASYMPTOTIC RELATIVE EFFICIENCIES	135

Chapter 1: Introduction

Today's world is a world of wide-area networks — Internet commerce, electronic mail, and the worldwide web are ubiquitous. Understanding the nature of the traffic traversing wide-area networks (WANs) is critical to ensuring the successful performance and operation of future incarnations of the network. Studying the nature of this traffic is often confounded by traffic that is abnormal or atypical in some regard.

Prior studies of WAN traffic have dealt with the problem of atypical traffic in an ad-hoc manner — traffic was uncovered and removed as the models were formulated. This thesis presents an algorithm to automate atypical traffic detection attempting to remove this ad-hoc nature of contaminate detection.

The bimodal nature of WAN traffic has also been uncovered in prior modeling efforts. This work reviews conventional techniques for modeling bimodal network traffic and formulates an alternative technique. These techniques are studied in general and in the context of the atypical traffic problem.

1.1. *Organization*

This thesis formulates and explores two techniques to deal with the problems atypical traffic presents to modeling efforts, and reviews the characteristics of atypical traffic and how it complicates these modeling efforts. The relevant background also is reviewed to provide a basis for the methods developed and criteria used to evaluate them. The thesis is organized as follows.

Chapter 2 reviews prior research characterizing WAN traffic, including the phenomena of bimodality, and highlights the various methods used to model WAN traffic. Chapter 2 then presents relevant background from the field of statistics, namely parameter estimation, measuring the quality of a model's fit, and robust estimation.

Chapter 3 reviews a previously studied, publicly available traffic archive exploring the nature, causes, and characteristics of atypical WAN traffic. The confounding effect of atypical traffic on probabilistic models is explored.

Chapter 4 formulates a parameterized algorithm for automated detection of atypical traffic from a collection of observations. The algorithm is evaluated using real

and simulated traffic, and the results are presented. This evaluation focuses on performance, scalability, and parametric sensitivity.

Chapter 5 reviews conventional techniques for parameter estimation of a class of bimodal distributions, the Gaussian mixture. A novel technique, the midpoint-distance profile, based on an extension of the concepts from robust statistics presented in Chapter 2, is developed. The performance of these estimation techniques is evaluated using simulation and conceptualized using real WAN traffic.

Chapter 6 summarizes the work presented and proffers several tracts for further study, including possible improvements to the algorithm of Chapter 4 and extensions to the estimation technique of Chapter 5.

Chapter 2: Background

Reliable prediction of telecommunication system performance depends upon an accurate understanding of the load placed on the networks. A traffic model mathematically describes the random effect of a user, or set of users, on resources within a communication system. Traffic modeling techniques for WAN traffic are varied. [1,2]

A commonality among all tracts of modeling is the reliance on certain statistical concepts, such as parameter estimation and goodness-of-fit. Parameter estimation involves determining the parameters of a model from a given dataset; the notion of goodness-of-fit is concerned with the discrepancy between a model and data on which that model is based.

This chapter begins with a review of several common modeling techniques for telecommunications traffic. A review of the theory of parameter estimation is presented, followed by a study of the current practice in assessing goodness-of-fit. Finally, some concepts from the field of robust statistics are presented.

2.1. *Network Traffic Modeling Techniques*

An overview of three different modeling techniques follows. By no means are these three techniques exhaustive — other techniques can be found in the literature. [1] However, the three techniques discussed facilitate further study of atypical network traffic. These techniques are empirical models, empirically derived analytic models, and self-similar models.

2.1.1. Empirical Models of WAN Traffic

Empirical models equate a set of observed data with the underlying probabilistic event. Consider an example of determining the height of a randomly selected person. The random variable to be modeled is height. Obtaining a set of people, measuring each one's height and equating that observation with the distribution of the random variable would formulate an empirical model. Continuing the example, if one measures ten people's heights and develops a model of height that model would predict the height of any person

(including people outside the observed set) as one of the ten observed values, with equal likelihood.

Initial work in formulating traffic models was done in an analogous manner. [3,4] By analyzing over 12,000,000 TCP packets, the authors were able to empirically model TCP conversation, or connection, statistics for a variety of application layer protocols, such as FTP, NNTP, and telnet. The method of modeling has a drawback — it is often quite complex to utilize in a simulation. As more data is observed the model becomes more accurate, but it becomes increasingly difficult to obtain a random sample within a simulation. Using empirically derived analytic models solves this problem of having model fidelity and accessibility fundamentally at odds with each other.

2.1.2. Empirically Derived Analytic Models

Analytic models use a known probability distribution function (PDF) to model a random variable. Returning to our height example, an analytic model would, for example, assume that the height of a person is distributed according to the normal, or Gaussian, distribution. A prediction from that model could be obtained by generating a random number adhering to the Gaussian distribution. Empirically derived analytic models base the parameters of the analytic distribution on observed statistics. Using the height example, an empirically derived analytic model would use the Gaussian distribution with a mean and standard deviation based on the 10 measured heights from the sample.

Network traffic models have been studied from this perspective. [5] Based on 3 million TCP connections observed over a three-year period, Paxson developed analytic models for the FTP, NNTP and telnet protocols. These analytic models are based on various known statistical distributions, such as the log-normal and the log-extreme distributions.

Both the empirical models and analytic models discussed have focused on TCP connections and have attempted to describe certain attributes of those connections, such as bytes transmitted, bytes received, or connection duration. Another tract of network traffic modeling involves looking at the traffic as more of a stochastic time-series. This approach has lead researchers to develop self-similar network traffic models.

2.1.3. Self-Similar Traffic Models

The seminal work of this tract of traffic modeling showed that the number of Ethernet packets observed on a network was self-similar. [6] Self-similarity is a property whereby a time-series is invariant against changes in scale, or time-aggregation. This is in sharp contrast to the putative Poisson process, used extensively in queuing theory. A Poisson process, at larger scales, will increasingly “smooth out”; whereas, a self-similar process will maintain patterns of peaks and valleys, despite the larger scale.

Since the initial work showing self-similarity in Ethernet traffic, many other researchers have found similar results. Intra-connection packet arrivals within certain TCP-based protocols, such as telnet, exhibit self-similar behavior. [7] Frame data generated by variable bit rate (VBR) video services were also found to be self-similar. [8] Generating time-series adhering to a self-similar model has proven non-trivial and synthesizing self-similar traces is an active research area. [9]

Self-similar models of network traffic subsume atypical traffic, i.e., when looking at Ethernet packet arrivals, all traffic, typical or atypical, is considered. This makes a certain degree of sense — if one’s goal is capacity planning, then all traffic has an impact, typical or not, and it should be modeled as it has an impact on the resources of the network. However, if one’s goal is to model the behavior of an application protocol, such as FTP, then atypical traffic obfuscates the modeling process. This thesis is concerned with problems akin to the later situation; thus, the implications and trappings of self-similarity are rendered moot.

2.2. *Bimodal Network Traffic*

Network traffic is often bimodal — consisting of two localized regions of relatively high probability densities. The bimodality of traffic often affects the analysis and design of new networks. Modeling bimodal traffic, therefore, is an integral part of modeling network traffic in general.

Message passing within a parallel processing environment serves as an example. One study of traffic in a parallel program benchmark found that 55% of the total bytes communicated were attributed to 98% of all the messages with a size of less than 40 bytes; whereas, 45% of the bytes communicated belonged to the 2% of messages 40

bytes or longer. [10] These findings were utilized in the analysis and design of new network interface architectures for message passing. [11]

Bimodality is not limited to message passing in parallel processing. The interarrival times of traffic in ATM networks has been found to be bimodal. [12] An additional example of bimodality can be found in the analysis and design of ring-based access protocols. [13,14]

The ubiquity of bimodality in network traffic underscores the importance of statistical techniques for estimation and modeling of bimodal distribution functions. Moreover, after exploring the characteristics of atypical WAN traffic in Chapter 3, we shall see that an extensible and robust estimation technique for multi-modal distributions can be tantamount to detection of atypical traffic.

A common distribution used to model bimodal variates is the Gaussian mixture. [15] This distribution represents a random variable that is drawn from one Gaussian component with a certain probability or weight and drawn from another Gaussian component the remainder of the time. This distribution can be generalized to more than two components. The probability distribution function for an n -component Gaussian mixture is given in Equation 2.1.

$$f_z(z) = \sum_{i=1}^n \frac{w_i}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(z-\mu_i)^2}{2\sigma_i^2}\right) \quad (2.1)$$

$$\sum_{i=1}^n w_i = 1$$

Depending on the values of the parameters, the two component Gaussian mixture can be multimodal. If the criterion of Equation 2.2 is met, the mixture cannot be bimodal. [15] Equation 2.2 describes the situation where the component means are sufficiently close, in terms of the component scale, for the components to overlap and cause the overall distribution to be unimodal.

$$(\mu_1 - \mu_2)^2 < \frac{27\sigma_1^2\sigma_2^2}{4(\sigma_1^2 + \sigma_2^2)} \quad (2.2)$$

However, if the criterion of Equation 2.3 is met, the distribution may be bimodal for certain values of w_1 and w_2 . [15] Equation 2.3 describes the situation where the

component means are sufficiently far apart, in terms of their respective scales, for the distribution to potentially be bimodal.

$$(\mu_1 - \mu_2)^2 > \frac{8\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad (2.3)$$

2.3. Parameter Estimation

Prior sections have touched on the importance of estimating and modeling network traffic parameters as analytically described random variables. Whether bimodal or not, analytic models require parameter estimation from an observed data set. This section begins with a discussion of some properties of estimators, then presents an overview of the statistical theory behind two estimation techniques: maximum likelihood estimation and the method of moments. Both of these methods are concerned with estimating the parameters of the probability density function of a random variable given a set of observations of that random variable.

2.3.1. Estimator Properties

Let $\hat{\theta}(\tilde{Z})$ be a function of a data set, $\tilde{Z}$, used to estimate the parameter θ of the underlying probability density of Z . Using this notation, three properties of estimators can be discussed: unbiasedness, consistency, and efficiency.

An estimator is unbiased if the expected value of the estimator is the true value of the parameter, as shown in Equation 2.4. [16,17]

$$E[\hat{\theta}(\tilde{Z})] = \theta \quad (2.4)$$

Unbiased estimators are desired since they provide an accurate estimate of the desired parameter and can be used in numerous circumstances with assurances of similar accuracy.

Consistent estimators converge to the true value as the sample size, m , increases. [16,17] This relationship is expressed in Equation 2.5.

$$\lim_{m \rightarrow \infty} P[|\hat{\theta}(\tilde{Z}) - \theta| \geq \varepsilon] = 0 \quad (2.5)$$

Consistent estimators are desirable as they provide a clearer picture of reality as more observations of a process are obtained.

Estimators of a random variable are themselves random variables. As such, estimators have variances. In general, the standard deviation of an estimator will decrease exponentially with the sample size, i.e., $\sigma_{\hat{\theta}} \propto m^b$. Most estimators have a rate of convergence of $b = -\frac{1}{2}$, the standard rate of convergence. [18]

The Cramer-Rao limit gives the best-case variance of the estimator, as shown in Equation 2.6. [17]

$$\text{var}(\hat{\theta}) \geq \frac{1}{I_f} \quad (2.6)$$

The term I_f is called the Fisher information and is given by Equation 2.7.

$$I_f = E \left[\frac{\partial \theta}{\partial \theta} \ln f(x|\theta) \right]^2 \quad (2.7)$$

An estimator is said to be efficient if it achieves the Cramer-Rao bound for all sample sizes.

In general, estimators are not efficient. However, the concept of relative efficiency can be used to compare to different estimators, $\hat{\theta}'$ and $\hat{\theta}''$. The relative efficiency of $\hat{\theta}'$ with respect to $\hat{\theta}''$ is given by Equation 2.8. (The factor of $\sqrt{m}$ is used to account for different sample sizes.)

$$e_{\hat{\theta}', \hat{\theta}''} = \frac{\text{var}[\sqrt{m}\hat{\theta}'']}{\text{var}[\sqrt{m}\hat{\theta}']} \quad (2.8)$$

Generally, relative efficiencies are referred to in the asymptotic case, i.e., $m \rightarrow \infty$.

2.3.2. Maximum Likelihood Estimation

The most widely used method of parameter estimation is the maximum likelihood estimation (MLE) technique. In this section, the theory behind maximum likelihood estimation is presented.

Given a probability density function for a continuous random variable, $f_z(z)$, and a sample of size m , $\tilde{Z} = \{z_1 \cdots z_m\}$, assumed to follow that distribution, the parameters of the distribution, $\bar{\theta}_z$, that are the most probable are given by the maximum likelihood estimator. [16,19] For a parameter space, $\bar{\theta}$, the maximum likelihood estimates, $\hat{\theta}_{MLE}$, are defined by the maximization shown below.

$$\begin{aligned}\hat{\theta}_{MLE} &= \arg \max_{\bar{\theta}} L(\bar{\theta}, \tilde{Z}) \\ L(\bar{\theta}, \tilde{Z}) &= \prod_{i=1}^m f_Z(z_i | \bar{\theta})\end{aligned}\tag{2.9}$$

The maximization problem described in Equation 2.9 is often transformed into the following using the log-likelihood function.

$$\begin{aligned}\hat{\theta}_{MLE} &= \arg \max_{\bar{\theta}} \ln(L(\bar{\theta}, \tilde{Z})) \\ &= \arg \max_{\bar{\theta}} \sum_{i=1}^m \ln(f_Z(z_i | \bar{\theta}))\end{aligned}\tag{2.10}$$

Furthermore, the maximization problem described in Equation 2.10 is often transformed into the minimization problem described by:

$$\hat{\theta}_{MLE} = \arg \min_{\bar{\theta}} - \sum_{i=1}^m \ln(f_Z(z_i | \bar{\theta}))\tag{2.11}$$

The transformation of Equation 2.9 into Equation 2.11 is done to exploit standard numerical routines for minimization problems. Therefore, a general procedure for solving for $\hat{\theta}_{MLE}$ is to numerically search for the minimum of the negative log-likelihood function presented in Equation 2.11.

Unbiased maximum likelihood estimators enjoy the property of asymptotically efficiency, at least in the single parameter case. [20] Equation 2.12 describes this behavior.

$$\lim_{m \rightarrow \infty} \text{var}[\sqrt{m}\hat{\theta}_{MLE}] = \frac{1}{I_f}\tag{2.12}$$

2.3.3. Method of Moments

Another technique for estimating the parameters of a distribution is the method of moments. [16,19] Consider a continuous random variable with a probability density function, $f_Z(z)$. The n-th moment of the distribution is given by:

$$M_n = E[Z^n] = \int_{-\infty}^{\infty} z^n f_Z(z) dz\tag{2.13}$$

For a given observation of the random variable, $\tilde{Z} = \{z_1 \cdots z_m\}$, the moment is estimated by:

$$\hat{M}_n = \frac{1}{m} \sum_{i=1}^m z_i^n \quad (2.14)$$

For a density function with N parameters, the expression of the first N moments can be found from Equation 2.13, in terms of the parameters, $\bar{\theta}_z$. Estimates of those moments can be found from Equation 2.14. By equating the analytic moments with their estimates, $M_n \equiv \hat{M}_n \quad \forall n \in \{1 \cdots N\}$, a system of N equations with N unknowns can be found. An estimate of the parameters of the distribution, $\hat{\theta}_{MOM}$, can be found by solving this system of equations.

An analogous formulation of the method of moments uses the central moments of a distribution, M'_n , which are defined by:

$$\begin{aligned} M'_1 &= M_1 \\ M'_n &= \int_{-\infty}^{\infty} (z - M'_1)^n f_Z(z) dz \quad n \neq 1 \end{aligned} \quad (2.15)$$

The estimates of the central moments are defined by:

$$\begin{aligned} \hat{M}'_1 &= \hat{M}_1 \\ \hat{M}'_n &= \frac{1}{m} \sum_{i=1}^m (z_i - \hat{M}'_1)^n \quad n \neq 1 \end{aligned} \quad (2.16)$$

The rest of the formulation of the method of (central) moments follows exactly as the method of moments.

The method of moments has the advantage of general applicability. However, it can generate biased and relatively inefficient estimators. [16]

2.4. *Measuring Quality of Fit*

In the previous section we reviewed the concept of parameter estimation. In this section, we review another important statistical concept — the problem of measuring the quality of fit of an analytic distribution to a set of observed data.

In general, the quality of fit problem is concerned with assessing, in a statistically significant manner, the deviation of an observation set from the distribution function hypothesized to describe that observation. Two standard methods, the chi-square and Kolmogorov-Smirnov tests, are well documented. [21,22,23] Another technique used in a

prior network traffic study [5] is the Pederson-Johnson estimate of model discrepancy. [24]

The Pederson-Johnson model discrepancy estimate is based on the chi-square estimate. The chi-square estimate relies on “lumped” data, e.g., for a continuous distribution the data are discretized into bins. [21,22,23] Given m observations of a random variable Z , the observations can be partitioned into N bins. Let $\tilde{Z}_i$ be the number of observations within the i^{th} bin. The chi-square test is based on comparing the observed bin observations against the expected value of the bin count. The expected value of the bin count, E , is given by:

$$E_i = mP[z \in i] = m \int_i f_z(z) dz \quad (2.17)$$

The chi-squared test is based on a measure of the difference between the expected and observed bin counts, namely:

$$X^2 = \sum_{i=1}^N \frac{(\tilde{Z}_i - E_i)^2}{E_i} \quad (2.18)$$

The primary shortcoming of the chi-square metric, $\chi^2 = X^2/m$, is that it cannot be used to compare fits where the metric is calculated using different numbers of bins, N . The Pederson-Johnson metric, λ^2 , eliminates this problem. Moreover, an estimate of the standard deviation of the estimate, $\hat{\sigma}_{\lambda^2}$, can be found. This allows statistically significant model comparisons, e.g., distribution “a” models the data better than distribution “b.” The Pederson-Johnson metric is estimated via:

$$\hat{\lambda}^2 = \frac{X^2 - K - df}{m - 1} \quad (2.19)$$

$$K = \sum_{i=1}^N \frac{\tilde{Z}_i - E_i}{E_i}$$

The term df in Equation 2.19 represents the degrees-of-freedom, which is given by $df = N - 1 - \Theta(\hat{\theta}_z)$, where $\Theta(\hat{\theta}_z)$ is the number of parameters of the distribution estimated from the observation. An estimate of the standard deviation of the estimate is given by:

$$\hat{\sigma}_{\lambda^2} = \sqrt{\hat{v}(\hat{\lambda}^2)}$$

$$\hat{v}(\hat{\lambda}^2) = \frac{2df + 4m\hat{\lambda}^2 + 4m\hat{\lambda}^4 + 4T}{m^2} \quad (2.20)$$

$$T = \sum_{i=1}^N \frac{D_i^3 - 2D_iE_i + \frac{5}{2}D_i^2 + \frac{3}{2}(D_i + E_i)}{E_i^2}$$

The remaining problem of this metric is determining the number of bins, N , to use in the calculation. To minimize the mean-square error in approximating a distribution with fixed size bins, the bin width, w , should be: [25]

$$w = 3.49\hat{\sigma}_z m^{-1/3} \quad (2.21)$$

Equation 2.21 provides a guideline for the number of bins to use for a given observation. A final refinement to tests of this sort, recommended by Jain [16], is that non-uniform distributions be transformed into a uniform distribution function using the cumulative distribution function:

$$\begin{aligned} \tilde{Z}_{Uniform} &= F_Z(\tilde{Z}) \\ F_Z(z) &= \int_{-\infty}^z f_Z(z) dz \end{aligned} \quad (2.22)$$

Based on the Pederson-Johnson model discrepancy estimate, comparisons of two or more hypothesized models can be made. This feature will be used in Chapter 3 when analyzing the effect of atypical WAN traffic on analytic models.

2.5. Robust Estimation

The previous sections have discussed parameter estimation and measuring goodness-of-fit, statistical concepts that require assumptions about the distribution of a random variable. The field of robust statistics is concerned with observations that deviate from those assumptions. [26] Concepts such as inlier versus outlier characterization influence of outliers on estimators and estimator breakdown fall within the purview of robust statistics.

The application of robust statistics is not new to network analysis — techniques from robust statistics have been applied to data mining to extract user profiling based on

web access logs. [27] In this section, two estimators from robust statistics, the midpoint of the shortest half (shorth) and the length of the shorth, are presented. These estimators are enhanced in this work to develop an extensible estimation procedure for multimodal distributions, as discussed in Chapter 5.

2.5.1. Midpoint of the Shortest Half

The midpoint of the shortest half is an estimator of location, just like the mean and median. [28] Analogous to the mean, which minimizes the mean of the squared residuals, the midpoint of the shortest half minimizes the median of the squared residuals. [29]

The midpoint of the shortest half can be seen as the midpoint of the range of data that is most closely grouped together. The method for obtaining the midpoint of the shortest half, $\hat{\theta}_{MPSH}$, begins with an observation set, Z , of size m . The data is ordered, to obtain, Z_o , defined by $Z_o = \{z_{[1]}, z_{[2]}, \dots, z_{[m]}\}$. The minimum distance corresponds to the most densely clustered half of the ordered set, Z_o :

$$d_{\min} = \min_{j \in \{1 \dots m-v+1\}} d_j$$

$$d_j = z_{[j+v-1]} - z_{[j]}$$

$$v = \left\lfloor \frac{m}{2} \right\rfloor + 1$$
(2.23)

The midpoint of the subset of data indicated by the minimum distance is the midpoint of the shortest half, or the least median of squares (LMS) estimator of location:

$$\hat{\theta}_{MPSH} = \frac{1}{2} (z_{[k-v-1]} - z_{[k]})$$

$$k \therefore d_k = d_{\min}$$
(2.24)

One drawback to the midpoint of the shortest half is that it does not enjoy the standard rate of convergence. Rather, the variance of estimates decreases as $m^{-1/3}$. [18] Therefore, the estimator is relatively inefficient.

2.5.2. Length of the Shorth

The length of the shorth [18] measures the range or spread of an observation set, analogous to the standard deviation, defined by:

$$\hat{\sigma}_z^2 = \frac{1}{m-1} \sum_{i=1}^m (z_i - \hat{\bar{z}})^2 \quad (2.25)$$

$$\hat{\bar{z}} = \frac{1}{m} \sum_{i=1}^m z_i$$

Unlike the standard deviation and other scale estimators, such as the median absolute deviation, the length of the shorth estimate does not require an estimate of the location, e.g., $\hat{\bar{z}}$, to estimate the scale of a data set. [30]

The length of the shorth, or location free scale estimate, is based on the distance of the shortest half, $d_{\min}$, from Equation 2.23. The estimate is scaled version of the distance of the shortest half:

$$\hat{\theta}_{LFS} = c' d_{\min} \quad (2.26)$$

$$d_{\min} = \min_{j \in \{1 \dots m-v+1\}} z_{[j+v-1]} - z_{[j]}$$

Where the constant, c' , depends on the particular distribution under study. For the normal distribution, c' is chosen so that $\hat{\theta}_{LFS}$ equals the standard deviation. The shortest half of a normal distribution, or the region of highest probability density, is centered about the mean. Therefore, half the distance of the shortest half corresponds to the 75th percentile. Using the inverse cumulative distribution function for the unit normal distribution:

$$\Phi^{-1}(p) = z \Rightarrow \Phi(z) = p \quad (2.27)$$

$$\Phi(z) = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$$

The constant, c' , can be chosen so that the location free scale estimate estimates the scale parameter of the normal distribution, i.e., the standard deviation:

$$\begin{aligned}\hat{\theta}_{LFS} &\equiv \hat{\sigma} \rightarrow d_{\min} = 2\sigma\Phi^{-1}(0.75) \\ \hat{\theta}_{LFS} &= c' d_{\min} \rightarrow c' = \frac{1}{2\Phi^{-1}(0.75)} \\ \hat{\theta}_{LFS} &\equiv \hat{\sigma} \rightarrow c' = 0.7413\end{aligned}\tag{2.28}$$

Using analogous logic, c' can be determined for an arbitrary distribution.

Unlike the midpoint of the shortest half, the length of the shorth converges at the standard rate, $m^{-1/2}$. [18] Therefore, estimators in this vein are expected to be fairly efficient.

2.6. Summary

This chapter presented an overview of some trends in network traffic modeling, including empirical models, empirically derived analytic models, and the bimodality of traffic. Additionally, statistical concepts, such as parameter estimation, measuring goodness-of-fit, and robust statistics, were reviewed.

The next chapter explores the effect of atypical traffic on modeling efforts. To quantitatively assess the confounding effect of contamination on modeling efforts, the concepts of goodness-of-fit presented in the chapter is utilized. After characterizing the effect of atypical traffic on modeling efforts in Chapter 3, Chapter 4 develops a technique for detection of atypical traffic.

Given the bimodal nature of some network traffic, Chapter 5 explores multimodal estimation. Extending the concepts from robust statistics presented in this chapter, the midpoint of the shortest half and the length of the shorth, a new method for multi-modal estimation is developed. This estimation technique is compared to the method of moments and maximum likelihood estimation.

Chapter 3: The Nature of Atypical Traffic

In the previous chapter, methods for modeling WAN traffic were reviewed (Section 2.1). In prior studies directed at both empirical models and empirically derived analytic models, atypical WAN traffic was removed from the observations prior to traffic modeling. This chapter focuses on atypical traffic: what is it, what causes it, how does it manifest itself, and what effects does it have on traffic models.

3.1. *Definition of Atypical Traffic*

A goal of modeling WAN traffic is to derive a model that applies in situations other than the one in which the traffic was observed. Ideally, a model of HTTP traffic, for example, would have a tunable “usage” parameter allowing simulation studies of networks under arbitrary load. Every network has idiosyncrasies. These idiosyncrasies are unique to a particular observation and do not represent the normal operation of the studied quantity. These idiosyncratic situations give rise to atypical traffic. If the modeling goal is generalness, atypical traffic should be removed. In the next section, the sources of atypical traffic are explored.

3.2. *Sources of Atypical Traffic*

Prior network traffic modeling efforts have experienced the problem of atypical traffic. [3,31] Based on those experiences, a taxonomy of atypical traffic sources can be constructed. Sources of atypical traffic exposed in prior studies fall into three categories:

- Abnormal, or malfunctioning, user-level applications that create abnormal connection statistics for the underlying TCP/IP protocol.
- Abnormal situations causing errors within a particular TCP/IP application protocol.
- Site-specific topologies or configurations that cause protocols to behave atypically.

Examples from the literature of each type are discussed in the following sections.

3.2.1. Abnormal User-Level Applications

Prior network studies have seen abnormal user-level applications in the form of periodic network traffic. [31] For example, a telnet script continually and regularly attempted to connect to a remote library catalog due to a hardware failure on the remote machine. The telnet connections associated with this situation displayed an irregularly high degree of deterministic properties for random network traffic. These properties included the hosts involved in the connections were always the same, the originating host always sent 3 bytes, the reply was always 175 bytes, and 35% of the connections arrived 180.43 seconds apart. Other examples of periodic traffic from the same study include finger requests arriving every 20 seconds due to a user running a background script and a substantial increase in ftp traffic due to a user's background process to download weather maps.

The validity of these uses for each application protocol is not in question; but the generalness is. In each of these three cases, protocol users were performing a task outside the scope of the random and typical operation of the protocol. A study attempting to generalize ftp data transfers would be thwarted by a particular user using ftp in a regularly recurrent manner. Therefore, these situations constitute abnormal user-level applications; therefore, the network traffic they generate is considered atypical.

3.2.2. Intrinsic Protocol Errors

The network news protocol (NNTP) [32] provides for the transfer of message postings associated with a particular group among servers. News servers throughout the Internet transfer messages between each other using NNTP. Previous traffic studies [3,31] observed a high number of constant transfers of 6 bytes. One study attributed this to server implementation errors and excluded those observations from further study. [3] Another study more thoroughly explained the situation prior to removing the connections further study — these connections corresponded to an NNTP server contacting a peer and being told that the remote peer was unable to communicate. The originating host then replies with the appropriate NNTP command and terminates the connection. This behavior results in 6 bytes being transferred from the originating host. [31]

Similar behavior was observed with the simple mail transfer protocol (SMTP). SMTP is a protocol for transferring electronic mail throughout the Internet. [33] Most SMTP connections transferred over 300 bytes; those that did not exhibited a sporadic distribution of bytes transferred. [31] The hypothesis presented was that the first 300 bytes were overhead and that connections with smaller transfers were failures, possibly due to busy remote servers or incorrect email addresses. Connections of this class were excluded from further study and characterization.

Another analogous situation can occur within the hypertext transfer protocol (HTTP). The HTTP specification defines several error conditions. [34] Two very common errors are “Location Not Found” (404) and “Internal Server Error” (500) that can occur, respectively, when an unknown entity is requested or when the server encounters an error servicing a request. HTTP servers, such as Apache [35] typically sent a static error page to the client when one of these situations is encountered. These error pages typically consist of a constant, or near constant, number of bytes. The presence of 404 errors in a server log is commonplace; recently, the propagation of the Code Red [36] and Nimda [37] worms tends to dramatically increase the number of 404 errors encountered by web servers.

All three of these protocols, NNTP, SMTP, and HTTP, have situations outside the scope of normal operation. In most cases, when these situations are encountered, the resultant connection consists of constant, or near constant, data transfer from either the originating host or the contacted host. As with abnormal user level applications, these connections do not represent typical protocol behavior; therefore they constitute another source of atypical traffic and have been excluded from characterization in prior studies. [3,31] In the next section, a final source of atypical traffic, site-specific network configuration, is explored.

3.2.3. Site Specific Network Configuration

Another situation encountered with NNTP traffic involved co-located servers. [31] Within the observed network, two NNTP servers existed that frequently exchanged information between each other in a master-slave relationship. This was not a conventional configuration, so the authors flagged these connections and excluded them

from further study. These co-located NNTP servers are an example of a site-specific network configuration that leads to atypical WAN traffic.

3.3. *Effect of Atypical Traffic*

The previous section discussed three sources of atypical traffic: abnormal user applications, intrinsic protocol behavior, and site-specific networks. This section explores the effect of atypical connections on modeling WAN traffic. Using publicly available traces of WAN traffic [38], empirical distribution function (EDF) plots of connection duration, bytes transmitted from the originating host, and bytes received from the originating host were created for NNTP, SMTP, and telnet traffic. These plots are presented in Appendix A. Using the EDF plots, qualitative inferences about the effect of atypical traffic are presented. A quantitative evaluation of the effect of atypical traffic on modeling tractability is then presented.

3.3.1. Qualitative Effect of Atypical Traffic

The EDF plots for telnet traffic show several qualities. The telnet observations are heavily contaminated by atypical traffic; about 73.7% of the connections represent atypical traffic. The distribution of transmitted bytes has a large discontinuity corresponding to the indicative 3 byte transfer of the anomalous traffic, as shown in Figure 3-1. When the atypical traffic is removed, the EDF is much smoother.

The EDF plots of Appendix A reiterate these trends. The distribution of duration also has a discontinuity; however, the connections associated with that cluster are a subset of the data associated within the 3 byte transfer cluster. Moreover, atypical traffic (defined by the 3 byte transfer cluster) may lie outside the discontinuity of the duration EDF. Thus, the 3 byte cluster of the transfer is more indicative of the atypical traffic. Finally, we note that the EDFs of all three variables are significantly smoother after the atypical traffic is removed.

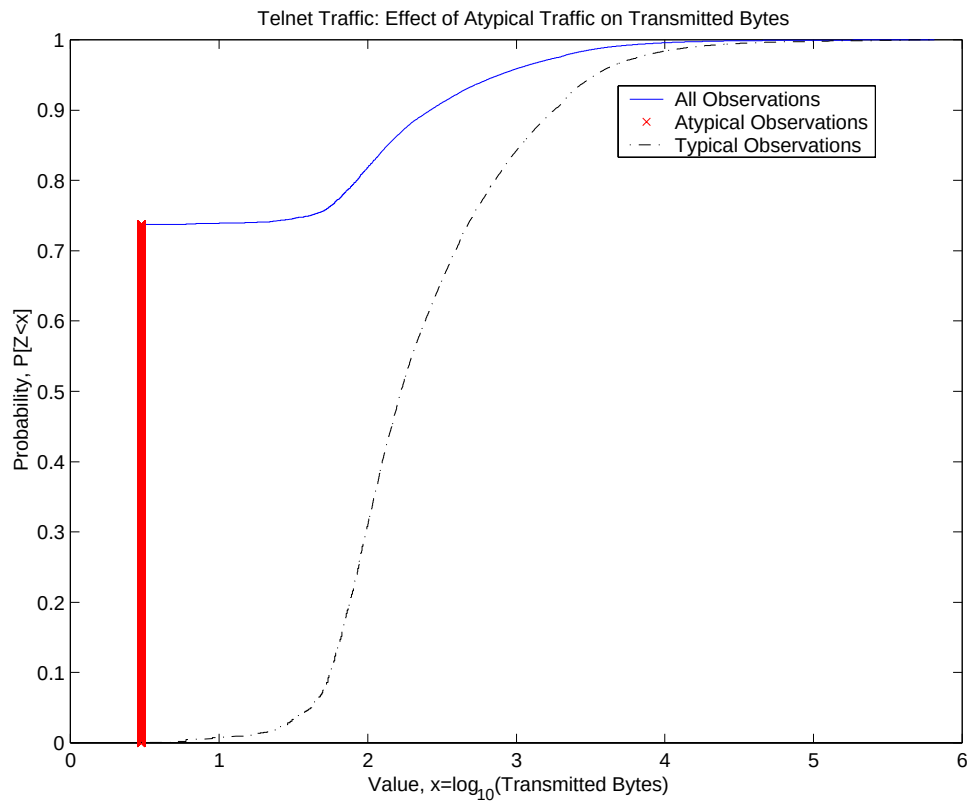


Figure 3-1 EDF of telnet transmitted bytes

The plots for SMTP data exhibit several differences. The level of contamination is small, about 1.2%. Also, the distribution of the duration and bytes received are approximately the same with and without the atypical traffic; therefore, the atypical behavior manifests itself only in the indicative variable, bytes transmitted. The indicative variable, bytes transmitted, does not show a discontinuity (i.e., a single value cluster), but rather a region associated with the contamination.

The plots comparing NNTP with and without contamination from atypical traffic are less dramatic. From the indicative transmitted bytes variable, it is apparently that the level of contamination is lower, about 4.5%. The distribution of received bytes also has a cluster, but that cluster represents a subset of the cluster associated with the indicative variable. For NNTP traffic, the “smoothness” of the distribution of remaining connections is not significantly improved over all connections; the EDF for the typical connections is similar to the EDF of all connections.

The effect of co-location on NNTP traffic is quite different. The distribution of duration for all connections is multimodal, indicated by multiple, disjoint regions of high slope. However, when the colocated observations are removed, the distribution becomes relatively smooth and single modal. The distribution of transmitted bytes, after removing the effect of collocation, is marginally “smoother” — one clustered region exists, rather than two. Also, the median of the distribution is shifted by an order of magnitude. The effect of collocation on received bytes is less noticeable. The median value is roughly the same; however, the proportion of low transfer connections is reduced. Qualitatively, the “smoothness” is not vastly increased by the remove of colocated connections.

Based on this qualitative analysis of the empirical distribution functions of these protocols, several inferences about the effect of atypical traffic can be drawn. Intrinsic protocol errors (i.e., as seen for NNTP) and abnormal user applications (i.e., as seen for telnet) have similar manifestations on connection variable distributions. These are noted below.

- These manifestations typically involve a single-valued cluster within an indicative variable, creating a discontinuity, or region of near infinite slope, on the EDF for that variable.
- The contamination described by the indicative variable often has analogous secondary manifestations on the other connection variables.
- The “height” of the indicative discontinuity is equal to the amount of contamination.
- The difference between the EDF of all and the EDF of typical connections is determined by the amount of contamination.

Whereas, anomalous traffic due to co-location has an entirely different set of manifestations:

- Co-located observations tended to have a multimodal duration. After the removal of colocated connections, the duration was unimodal, favoring the larger mode.
- The effect of co-location on byte transfer can vary. In one instance the median was increased an order of magnitude; in another it was unaltered.

3.3.2. Quantitative Effect of Contamination on Model Tractability

A technique for assessing goodness-of-fit was presented in Section 2.4. The model discrepancy, $\hat{\lambda}^2$, and its standard deviation, $\hat{\sigma}(\hat{\lambda}^2)$, were calculated for the data shown in Appendix A. The hypothesized model distribution was the Gaussian distribution. Perhaps other distributions more accurately model the data; however, by comparing model discrepancy for a fixed distribution, a measure of the model tractability can be obtained. The measure of tractability assumes that the better a simple distribution (e.g., the Gaussian) models the data the more easily the data can be modeled analytically. These results are shown in Table 3.1.

Table 3-1: Measuring Modeling Tractability Via Model Discrepancy

	Number of Observations, m	Duration		Bytes Transmitted		Bytes Received	
		$\hat{\lambda}^2$	$\hat{\sigma}(\hat{\lambda}^2)$	$\hat{\lambda}^2$	$\hat{\sigma}(\hat{\lambda}^2)$	$\hat{\lambda}^2$	$\hat{\sigma}(\hat{\lambda}^2)$
Telnet, All Connections	70137	9.71	0.1216	22.7	0.263	22.7	0.266
Telnet, Typical Connections	18443	0.0313	0.00278	0.1865	0.00771	0.0218	0.00218
NNTP, All Connections	130673	0.450	0.00480	0.944	0.00960	2.57	0.0243
NNTP, Non-error connections	124773	0.450	0.00509	1.005	0.01063	3.33	0.0329
NNTP, Typical Connections	65509	0.267	0.00501	1.151	0.01681	5.45	0.0785
SMTP, All Connections	205405	0.1391	0.001927	0.217	0.00246	1.402	0.01056
SMTP, Typical Connections	202966	0.1332	0.001900	0.230	0.00267	1.413	0.01075

These results confirm several inferences made in Section 3.3.1. The removal of the large number of atypical telnet traffic significantly decreased the modeling discrepancy for all three variables. However, the removal of the relatively small amount of NNTP and SMTP contamination had little effect on the model discrepancy, which was already relatively low. The increased modeling discrepancy of NNTP duration due to collocation is noticeable; however, removing collocated connections increased the other two variables' discrepancy.

3.4. Conclusions

This chapter defined atypical traffic as network traffic attributed to idiosyncratic network or protocol behavior that undermines a ubiquitous traffic model. The sources and

effect of these situations were explored. Two patterns of contamination, abnormal user applications and intrinsic protocol behaviors, caused similar effects. These effects included single-valued discontinuities of the empirical distribution function, primarily on the indicative variable, but also on the other connection variables. If the level of contamination is significant, the removal of atypical traffic dramatically increases modeling tractability. The utility of an algorithm for automatic detection of atypical traffic based on the EDF is evident from these observations. Chapter 4 presents an algorithm developed to address this need.

Another source of atypical traffic, a site-specific network topology, caused a multimodal distribution of connection duration. Removing the atypical traffic created a single mode distribution favoring the larger mode and increased the modeling tractability. The effect of a site-specific topology on the other variables was varied. From these observations, the need for a robust and extensible multimodal estimation technique is apparent. Chapter 5 presents an estimation technique meeting this need.

Chapter 4: Detection Algorithm for Atypical WAN Traffic

Prior chapters have addressed the problem atypical WAN traffic poses to modeling WAN traffic analytically. Characterizations of the nature and effect of atypical WAN traffic have also been presented. This chapter develops an algorithm for detection of atypical traffic from collected data beginning with a statement of the criteria and goals of the algorithm. Next, a thorough description of the algorithm is presented, followed by a description of the implementation of the algorithm in MATLAB. Finally, several aspects of the developed algorithm are studied, particularly for correctness, stability, parametric sensitivity, and scalability.

4.1. Algorithm Definition and Objectives

A general-purpose algorithm for the detection of atypical WAN traffic should be a *parameterized, automated technique to detect and formalize contamination models using only a set of observations and metadata describing those observations*. This definition, based on the previously discussed nature of atypical traffic, describes the algorithm presented below.

The algorithm should be automated to allow detection of atypical traffic without knowledge of the source the atypical traffic. For example, in the case of the failing telnet script, the root cause of the atypical traffic was determined prior to excluding the data. [31] The algorithm merely identifies potential contaminate situations on the basis of the connection properties, not the underlying cause.

The algorithm should be parameterized. The parameters associated with the algorithm should be indicative of the level and severity of the contamination in a generally applicable manner. Parameters should not presuppose a familiarity with the contamination specifics.

The algorithm should produce formalized contamination models. Rather than merely flagging potential atypical regions as a set of data indices, the technique should describe these regions based on the indicative connection variable. For example, rather than describing a contaminate region as “observations 1, 3, 4, and 13”, the region is

described as “bytes transmitted equals 3”. Observation metadata, such as the names of the variables, is required to facilitate this formalization.

In addition to satisfying this definition, a desirable detection algorithm should also meet the following objectives: correctness, stability, and minimality of hypothesized contamination models. Confirming that the algorithm, with sensibly selected parameters, detects atypical regions for a previously studied set of observations will show correctness. Stability requires that the algorithm converge to a specific set of hypothesized contamination models, for a given input and set of parameters. An ancillary objective is that the set of contaminate models be minimal, i.e., if one model is subsumed by another model only the subsuming model should be reported.

We shall proceed to describe the particulars of the algorithm, having described the nature and objectives of a detection algorithm for atypical WAN traffic. The next section begins that discussion by explaining some notation and terminology.

4.2. *Notation and Terminology*

The overall collection of observations being studied is called the dataset, D . D is a collection of m observations, each with n variables. Therefore, D , can be represented as $m \times n$ matrix of observations, taken row-wise:

$$D = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} \quad (4.1)$$

A working dataset, D^w , is similar to the dataset, D ; however, D^w may contain fewer than m observations. D^w will still contain n variables for each observation. D^w can also be represented in matrix form:

$$D^w = \begin{bmatrix} x'_{1,1} & \cdots & x'_{1,n} \\ \vdots & & \vdots \\ x'_{m^w,1} & \cdots & x'_{m^w,n} \end{bmatrix} \quad m^w \leq m \quad (4.2)$$

Each of the variables in D and D^w has some physical meaning. In the case of WAN connections the variables could represent connection duration, bytes transmitted,

and bytes received. The observational metadata, MD , describes each variable, x_i , of the dataset. The elements of MD for each variable are described in Table 4.1

Table 4-1 Metadata Elements

Metadata Element	Meaning	Possible Values
name	The name of the variable	E.g., “duration”, ”bytestx”
int	Denotes integer, rather than real valued, variables	0 or 1
log	Denotes variables typically modeled on a logarithmic scale	0 or 1

The set of contaminate, or atypical, regions extracted from D are called the set of accepted contamination models, C . Since models are formalized, elements of C are rules describing the contamination model’s relationship to the data, e.g., “bytestx = 3”. When referring to a set of detected models, prior to acceptance, the terminology used is hypothesized contamination models, denoted C^H .

The next section describes the algorithm for detection of atypical WAN traffic using the terminology presented in this section.

4.3. Algorithm Description

The algorithm for detecting atypical WAN traffic is summarized by Figure 4.1. The technique begins by defining the working dataset as all observations not associated with any currently accepted contamination models. Obviously, no contamination models have been detected at the outset, so the working dataset is equivalent to the original dataset. Section 4.3.1 describes this process in greater detail. Next, regions of contamination are detecting using a single variable of the working dataset. The technique employs a data windowing technique on the empirical distribution function, and is described in Section 4.3.2. The technique for single-variate contamination detection relies on two parameters, the minimum level of contamination, α , and the critical angle, θ_c , which is indicative of the severity of the clustering of the contamination. A larger value of θ_c will lead to the detection of only narrow “spikes” of contamination; a smaller value of θ_c will detect broader regions of contamination. This single-variate detection

technique is applied for all variables present in the working dataset and the resultant models are combined to form the set of hypothesized contamination models. If no contamination models are detected, the process terminates.

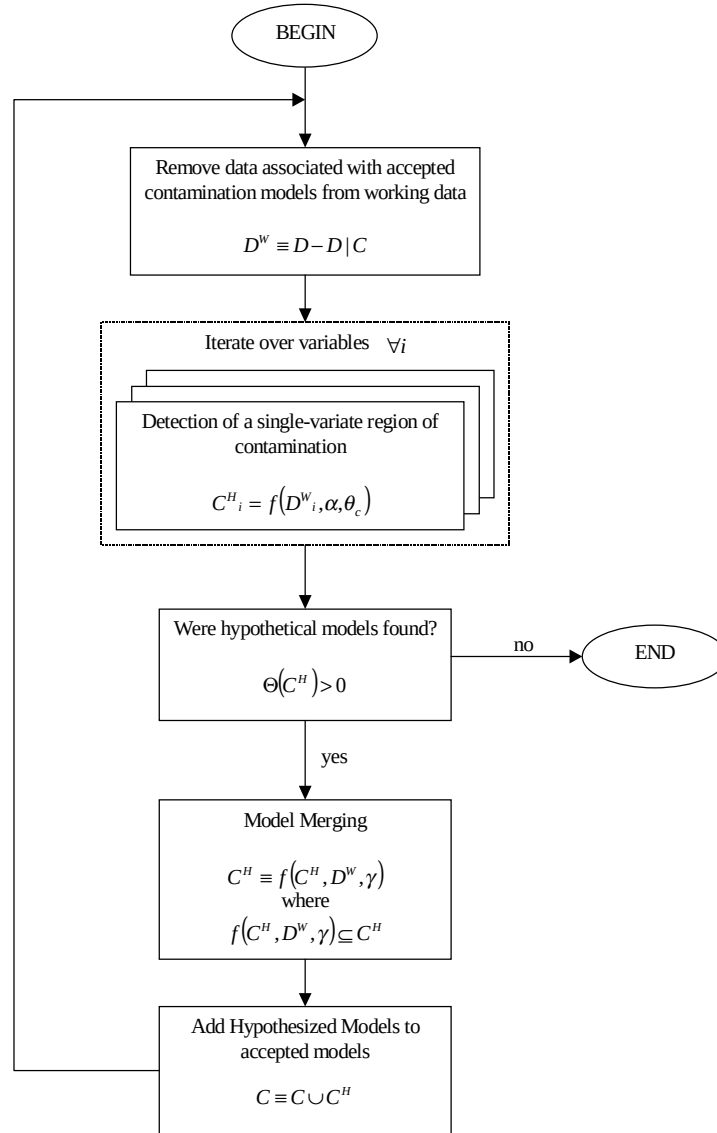


Figure 4-1 Detection Algorithm Overview

If the set of hypothesized contamination models is not empty, the models are merged to ensure minimality of models. The merging process is described in Section 4.3.3. This process relies on a parameter, the minimum overlap, γ , which is indicative of the amount of correlation between models required for one model to subsume another.

The manner in which model merging is done has implications to the global convergence of the algorithm. These implications are also discussed in Section 4.3.3.

Finally, the set of hypothesized models after merging is combined with the set of contamination models to form the new set of contamination models. The technique begins again, censoring the dataset to form the working dataset.

4.3.1. Censoring Data Based on Contamination Models

Prior to contamination detection, the set of data not associated with any existing contamination model, the working dataset, must be determined. This operation is expressed in the following equation:

$$D^w \equiv D - D | C \quad (4.3)$$

The set $D | C$ represents data elements associated with contamination models. The elements of $D | C$ can be found by evaluating the rules of each model, C_i , against the dataset, D . This evaluation will result in set of indices, κ_i , representing rows of the dataset to be excluded. The working dataset is obtained by excluding all of the rows of D associated with all the models of C . Therefore, Equation 4.3 can also be expressed as follows, where $\Theta(C)$ denotes the number of models in C .

$$D^w = D \left(\{1, \dots, i, \dots, m\} - \bigcup_{i=1}^{\Theta(C)} \kappa_i \right) \quad (4.4)$$

The next step in the algorithm involves the detection of single-variate regions of contamination. This step, which relies on the working dataset, is described in the following section.

4.3.2. Single-Variate Contamination Region Detection

Using the working dataset, as described above, the next process of the algorithm detects contamination regions associated with a single variable of the working dataset. As mentioned above, this process has two parameters, α , the minimum level of contamination to be detected and θ_c , the measure of contamination severity. The detection process results in a set of hypothesized contamination models and can be expressed as:

$$C^H_i = f(D^w_i, \alpha, \theta_c) \quad (4.5)$$

The single-variate contamination detection process, represented by the function f , is repeated for each variable of D^w and the set of hypothesized models, C^H , represents the union of the models detected for all variables. The entire process can be expressed as:

$$C^H = \bigcup_{i=1}^n C^H_i \quad (4.6)$$

$$C^H_i = f(D^w_i, \alpha, \theta_c) \quad \forall i \in \{1, \dots, n\}$$

The single-variate contamination detection function, f , is based on the characteristics of atypical traffic developed in Chapter 3. Chapter 3 established that the majority of atypical WAN traffic appears as a vertical, or near vertical, line on the empirical distribution function (EDF). This vertical “jump” on an EDF, or equivalently the “spike” on a histogram, represents data elements clustered around a single value. Detection of these regions relies on a process to uncover large concentrations of single, or near-single, valued observations. In general, the observed variable will be continuous, e.g. connection duration; therefore, simple methods based on the frequency of a particular observation would be insufficient.

The method developed uses regions of high slope of the EDF to detect clustered regions. Network traffic is often studied on a logarithmic scale; if the corresponding element of the metadata, MD , so indicates, the data will be transformed to a logarithmic scale, e.g., $D^w_i \equiv \log_{10}(D^w_i)$, at the outset of the detection process.¹ Since a slope-based technique is used, the abscissa of the EDF should be normalized but, generally, the range of observations will not be standard or normalized. This abscissa normalization is accomplished by a simple scaling of the data into the interval $[0,1]$. After this normalization, the EDF is represented by a set of x, y data points, defined by:

¹ In general the base of the logarithm scale need not be 10 for the distribution of the data to be conducive to modeling. The base of 10 is chosen on the basis of its ubiquity.

$$x_j = \frac{D^w_{i[j]} - \min(D^w_i)}{\max(D^w_i) - \min(D^w_i)}$$

$$y_j = \frac{j}{\Theta(D^w_i)} \quad (4.7)$$

$$j \in \{1, \dots, \Theta(D^w_i)\}$$

Elements of the EDF are used in a detection process that is based on a growing window of data. Starting with the first observation, $j = 1$, the position of the data point defined by the positive alpha-percentile offset is calculated via:

$$\rho(j, \alpha) = \max(j + \lceil \alpha \Theta(x) \rceil, \Theta(x)) \quad (4.8)$$

The current window of data is composed of elements of the EDF between j and $\rho(j, \alpha)$. The window may indicate the presence of contamination on the basis of the slope of the region compared to the critical angle. The presence of contamination is determined via the test described by the inequality:

$$\theta_c < \tan^{-1} \left(\frac{y_{\rho(j, \alpha)} - y_j}{x_{\rho(j, \alpha)} - x_j} \right) \quad (4.9)$$

If the slope is large enough to indicate the presence of contamination, the window is grown until the condition of Equation 4.9 is not met. The elements of the working dataset, D^w_i , associated with the EDF window represent a hypothesized contamination model. This contaminate model is added to the set, C^H_i . The exact procedure whereby a window of EDF data is translated to a rule-based contamination models is termed *contamination model formalization* and will be discussed later in this section. The sliding window technique resumes, with a new window starting at the element immediate after the current window.

However, if the current window does not indicate the presence of contamination on the basis of Equation 4.9, the window is “slid” to the right one datum, e.g. $j \equiv j + 1$, and the test for contamination is repeated.

Figure 4-2 illustrates this sliding window technique. The ordered data are compared, calculating the angle between the current datum and the positive α -percentile offset datum. For the data of Figure 4-2, the 6th observation gives means the critical angle

criteria. At this initial stage, the data in the α -sized window comprise a region of contamination. (Data denoted as circles.) The algorithm grows the window until the critical angle criterion is not met. (The data point represented as a square.) The window is shrunk by one datum and the region is reported as contamination. (The circles and triangles.)

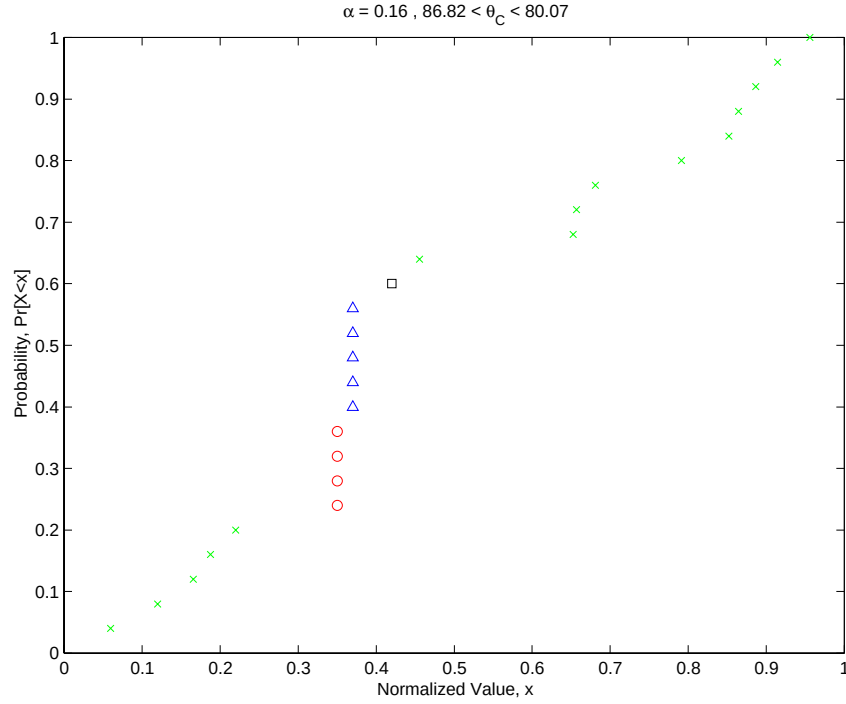


Figure 4-2 Single-variate detection illustration

The sliding window process is repeated until all data of the EDF have been examined. As a result of this technique, all regions of contamination with severity greater than θ_c and with more than α -percentage elements will be represented in C^H_i . The translation of a window of EDF indices into a rule-based contamination model has been alluded to in this discussion. The exact procedure of contamination model formalization will be discussed in the following section.

4.3.2.1. Contamination Model Formalization

The EDF slope-based sliding window detection algorithm described above determines a set of indices, λ , of the EDF of the single-variate working dataset, D^W_i .

The subset of the working data associated with the contamination is defined by the following where the notion $[\lambda]$ indicates indices of the ordered dataset and $\{\lambda\}$ indicates indices of the original (unordered) dataset.

$$D^C \equiv D^W_{i\{\lambda\}} \quad \lambda \ni D^W_{i\{\lambda\}} = D^W_{i[\lambda]} \quad (4.10)$$

Using the values of $\min(D^C)$ and $\max(D^C)$, and the observation metadata, MD , the region of contamination can be translated into a formal contamination rule. The particulars of this formalization process are straightforward. For example, if $\min(D^C) = \max(D^C) = 10$, the variable under consideration is termed “bytestx”, and the variable is flagged as integer-valued, then the formalized rule corresponding to D^C would be “bytestx = 10”.

4.3.3. Contaminate Model Merging

Given a set of contamination models, C , and a dataset, D , the possibility exists of correlation between elements of C . Given the goal of minimality of proposed contamination models, a process was developed to eliminate this correlation. This process, employing a matrix-reduction approach, is described below.

The coverage of contamination model C_i by model C_j , denoted $\psi_{i,j}$ represents the fraction of data in C_i that are also modeled by C_j . This coverage, denoted $\psi_{i,j}$, is defined by:

$$\psi_{i,j} = \frac{\Theta(C_j \cap C_i)}{\Theta(C_i)} \quad (4.11)$$

The coverage matrix, CM , is defined by calculating the coverage for each element of C . Noting that $\psi_{i,j} = 1$ and assuming that C has m elements, the coverage matrix takes the following form:

$$CM = \begin{bmatrix} 1 & \psi_{1,2} & \cdots & \psi_{1,m} \\ \psi_{2,1} & 1 & \cdots & \psi_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{m,1} & \psi_{m,2} & \cdots & 1 \end{bmatrix} \quad (4.12)$$

Augmenting the coverage matrix with the values of i yields the augmented coverage matrix, ACM , defined by:

$$ACM = \left[\begin{array}{cccc|c} 1 & \cdots & \psi_{1,j} & \cdots & \psi_{1,m} & 1 \\ \vdots & 1 & & & \vdots & \vdots \\ \psi_{i,1} & & \ddots & & \psi_{i,m} & i \\ \vdots & & & 1 & \vdots & \vdots \\ \psi_{m,1} & \cdots & \psi_{m,j} & \cdots & 1 & m \end{array} \right] \quad (4.13)$$

The augmented coverage matrix can be used to reduce the set of models, C , to the models having mutual coverage less than a threshold value, γ , using a matrix-reduction method. The largest off-diagonal element of the coverage matrix is located, $\psi_{i,j}^{\max}$. The i^{th} row and column of the augmented coverage matrix is removed if $\psi_{i,j}^{\max} > \gamma$. This represents the subsuming of C_i by C_j . This process continues until $\psi_{i,j}^{\max} < \gamma$. The last column of the augmented coverage matrix then gives the minimal set of contamination models. Appendix B shows an example of this merging process.

Figure 4.1 describes the model merging process via the equation:

$$C^H \equiv f(C^H, D^W, \gamma) \text{ where } f(C^H, D^W, \gamma) \subseteq C^H \quad (4.14)$$

The matrix reduction technique described above serves as the function, f , but the choice of arguments warrants discussion. Appendix B (Section B.2) demonstrates a potential problem with basing the model merging on all contamination models, rather than just the recently detected contamination models, e.g., $f(C, D, \gamma)$ instead of $f(C^H, D, \gamma)$. Given certain properties of the data, using $f(C, D, \gamma)$ in the model merging stage will cause a lack of convergence in the entire contamination detection algorithm. In short, a model will be detected but merged, only to be re-detected on the next iteration. This will continue indefinitely.

Furthermore, basing the model merging on all models but only the working data, e.g. $f(C, D^W, \gamma)$, is problematic. It is likely that some models in C will have no applicable data in D^W . This would cause the numerator and denominator of Equation 4.11 to be zero, leading to an indeterminate form.

Since merging of the form $f(C, D, \gamma)$ can lead to a divergent algorithm and merging of the form $f(C, D^W, \gamma)$ can lead to indeterminate forms, two candidates remain: $f(C^H, D, \gamma)$ or $f(C^H, D^W, \gamma)$. Appendix B (Section B.2) explores the

differences between these two choices. Merging based on $f(C^H, D, \gamma)$ will favor a minimal set of contamination models; consequently, some contamination may be unmodeled and remain in the data. Merging based on $f(C^H, D^W, \gamma)$ will favor a more thorough cleaning of the data at the expense of the minimality of hypothesized models. Also, merging based on $f(C^H, D^W, \gamma)$ is more efficient numerically, as the coverage matrix is calculated from a smaller number of elements. Given these differences, merging based on the working dataset was chosen to ensure that the resultant data is as decontaminated as possible.

4.3.4. Incorporation of C^H into C

After the detected models have been merged to eliminate any strong redundancy, the set of hypothesized models, C^H , are joined with the existing set of accepted models to form a new set of accepted models, $C \equiv C \cup C^H$. At this point, the algorithm begins a new iteration, attempting to detect any contamination models in the working dataset formed by excluded data associated with the newly formed set of accepted contamination models. The algorithm will continue until no contamination models are hypothesized, e.g. $C^H = \emptyset$.

4.4. Algorithm Implementation

The contamination detection technique described in Section 4.3 was implemented in MATLAB. Appendix C contains descriptions and source code for each of the routines utilized by the technique. The implementation of these routines was straightforward; however, one notable variation is evident in the source. This variation is highlighted below.

The sliding window technique for detecting regions of contamination based on a single variable described in Section 4.3.2 was altered slightly. The description in Section 4.3.2 implicates a nested iterative approach; however, MATLAB scripts perform significantly better if iterations can be implemented as vector operations. This was accomplished by calculating the slope between each datum and its α -percentile neighbor. This vector of results was then searched for contiguous regions greater than the

critical angle. These contiguous regions formed the set of regions of contamination to which the sliding window algorithm was applied.

4.5. Evaluation and Performance

In the previous two sections, the algorithm for contamination model extraction was described and its implementation was discussed. This section focuses on evaluating the algorithm and studying its performance. First the algorithm is evaluated for the criteria established in Section 4.1: correctness, stability, and model minimality. Then the performance of the technique is discussed, with particular attention paid to the effect of the parameters, α , θ_c , and γ . Additionally, the topic of scalability is explored.

4.5.1. Correctness

The test for correctness of the algorithm was established in Section 4.1. Using prior knowledge of a contaminated dataset and comparing that knowledge to the results of the contamination detection algorithm, the correctness can be established, albeit loosely. Due to the nature of defining contamination regions and the effect of the parameters, “correctness” is subject to relative performance comparisons. Subject to this caveat, the tests of Section 4.1 were conducted using the data studied in Chapter 3 and a randomly generated dataset to illustrate correctness.

Section 3.2.1 described the atypical traffic in the LBL telnet trace. Based on that analysis, the prevailing contamination model is “bytestx = 3”. The Listing 4-1 confirms that this model is extracted using the parameters: $\alpha = 0.1$, $\theta_c = 89.9^\circ$, and $\gamma = 0.9$.

```

>> Debug.edf = 0; Debug.trace = 1; Debug.pause = 0;
>> load TelnetData D MetaData
>> DetectedModels = process( D , 0.1 , 89.9 , 0.9 , MetaData , Debug );
Iteration 1

Hypothesized Models
[1] (bytestx == 3.000000)
[2] (175.000000 <= bytesrx) & (bytesrx <= 177.000000)
[3] (2.241387 <= log10(duration)) & (log10(duration) <= 2.243036)
[4] (2.243722 <= log10(duration)) & (log10(duration) <= 2.248432)

Coverage Matrix
1.00000 0.92515 0.18945 0.54206
0.99975 1.00000 0.19309 0.53228
0.99817 0.94140 1.00000 0.00000
0.99861 0.90743 0.00000 1.00000

Results of Model Merge
1

Accepted Models
[1] (bytestx == 3.000000)

```

Listing 4-1 Detection Trace for Telnet Data

Section 3.2.2 described the contamination of the NNTP traffic in the LBL trace. For the contamination detection technique to succeed, the model “bytestx = 6” should be extracted. The level of contamination in NNTP is low (4.5%), so the following parameters were used: $\alpha = 0.04$, $\theta_C = 89.9^\circ$, and $\gamma = 0.8$. Listing 4-2 confirms that this model was extracted, among others. Given the goal of detecting potential regions of contamination, without knowledge of the underlying network conditions, it is reasonable to expect the algorithm to over-detect in this manner.

```

>> Models = process( D , 0.04 , 89.9 , 0.8 , MetaData , Debug);
Iteration 1

Hypothesized Models
[1] (bytestx == 6.000000)
[2] (1199.000000 <= bytesrx) & (bytesrx <= 1201.000000)
[3] (1370.000000 <= bytesrx) & (bytesrx <= 1371.000000)
[4] (1559.000000 <= bytesrx) & (bytesrx <= 1560.000000)

Coverage Matrix
1.00000 0.00000 0.00000 0.00000
0.00000 1.00000 0.00000 0.00000
0.00000 0.00000 1.00000 0.00000
0.00000 0.00000 0.00000 1.00000

Results of Model Merge
1      2      3      4

Accepted Models
[1] (bytestx == 6.000000)
[2] (1199.000000 <= bytesrx) & (bytesrx <= 1201.000000)
[3] (1370.000000 <= bytesrx) & (bytesrx <= 1371.000000)
[4] (1559.000000 <= bytesrx) & (bytesrx <= 1560.000000)

```

Listing 4-2 Detection Trace for NNTP Data

```

Iteration 2

Hypothesized Models
[1]      (bytesrx == 120.000000)

Coverage Matrix
1.00000

Results of Model Merge
1

Accepted Models
[1]      (bytestx == 6.000000)
[2]      (1199.000000 <= bytesrx) & (bytesrx <= 1201.000000)
[3]      (1370.000000 <= bytesrx) & (bytesrx <= 1371.000000)
[4]      (1559.000000 <= bytesrx) & (bytesrx <= 1560.000000)
[5]      (bytesrx == 120.000000)

```

Listing 4-2 (continued)

Finally, a randomly generated data set was used. This dataset consisted of 10,000 observations of a three-variable data space. There were four randomly generated contamination models, each appearing with a randomly assigned probability; otherwise, the data was distributed normally. Listing 4-3 shows that, with the parameters $\alpha = 0.1$, $\theta_c = 89.99^\circ$, and $\gamma = 0.9$, only 3 of the models are correctly extracted. However, if $\alpha = 0.05$, all four models are detected.

```

>> Debug.edf = 0; Debug.trace = 1; Debug.pause = 0;
>> [D,MetaData,Models] = randomData( 4 , 3 , 10000 );
>> DetectedModels = process( D , 0.1 , 89.99 , 0.9 , MetaData , Debug );

Iteration 1

Hypothesized Models
[1]      (x1 == -3.000000)
[2]      (x2 == -3.000000)
[3]      (x3 == -1.000000)

Coverage Matrix
1.00000 0.59709 0.69098
0.68860 1.00000 0.68592
0.69229 0.59589 1.00000

Results of Model Merge
1      2      3

Accepted Models
[1]      (x1 == -3.000000)
[2]      (x2 == -3.000000)
[3]      (x3 == -1.000000)

>> printModels(Models)

[1]      x2 == 3
[2]      x2 == -3
[3]      x3 == -1
[4]      x1 == -3

>> DetectedModels = process( D , 0.05 , 89.99 , 0.9 , MetaData , Debug );

Iteration 1

```

Listing 4-3 Detection Trace for random data

```

Hypothesized Models
[1]      (x1 == -3.000000)
[2]      (x2 == -3.000000)
[3]      (x3 == -1.000000)

Coverage Matrix
1.00000 0.59709 0.69098
0.68860 1.00000 0.68592
0.69229 0.59589 1.00000

Results of Model Merge
1      2      3

Accepted Models
[1]      (x1 == -3.000000)
[2]      (x2 == -3.000000)
[3]      (x3 == -1.000000)

Iteration 2

Hypothesized Models
[1]      (x2 == 3.000000)

Coverage Matrix
1.00000

Results of Model Merge
1

Accepted Models
[1]      (x1 == -3.000000)
[2]      (x2 == -3.000000)
[3]      (x3 == -1.000000)
[4]      (x2 == 3.000000)

```

Listing 4-3 (continued)

The contamination detection algorithm extracted the prevailing contamination models for both the LBL Telnet traffic and the LBL NNTP traffic. This provides some confidence in the overall correctness of the technique. A correctness test also was done using a randomly generated dataset. This test also confirms the validity of the technique; however, it suggests the importance of the parameters in the success of the algorithm, a topic that will be addressed in Section 4.5.4.

4.5.2. Stability

Two conditions must be met for the overall stability of the algorithm. First, each sub-process must be stable and the overall algorithm must cease iterating by assuring that $C^H = \emptyset$ at some point. The second criteria can be assured only if model merging is based on only C^H and not C , as discussed in Section 4.3.3 and Appendix B. Alternatively, the overall algorithm can stop if all the data is associated with a contamination model. In either case, the stability of the overall iterative structure is assured.

Examining the nature of each component process can show that the first criterion holds, i.e. that each sub-process is stable. The first stage, removing contaminate associated data, is a set reduction using two finite sets; therefore, this process will be stable. The next stage, contaminate region detection involves nested iteration over two finite sets: the variables in the dataset and the observations of data. Again, this process will complete in finite time. Finally, model merging is a convergent process, as it reduces the size of its own operational set, the coverage matrix.

These two criteria, the stability of the overall algorithm flow and the stability of each component process, provide confidence that the contamination detection technique is stable; i.e., it will not continue indefinitely; rather, it will reach some ultimate and final stage.

4.5.3. Model Minimality

The final objective of the contamination model detection algorithm is to ensure minimality of set of contamination models extracted. The model merging stage, based on the coverage matrix, developed in Section 4.3.3 accomplishes this goal. Listing 4-1 demonstrates the utility of this technique, as four models are merged into one. As discussed in Section 4.3.3 and Appendix B, a tradeoff exists between the goal of model minimality and the “smoothness” of the final data. The technique developed favors data decontamination over model minimality.

4.5.4. Effect of Parameters on Contamination Detection

A review of Listing 4-3 (Section 4.5.1) reveals the importance of the three parameters, α , θ_c , and γ , on the overall success of the technique. The effect of these parameters is studied by reviewing their respective meanings in the context of the traces presented in Section 4.5.1. Finally, a simulation experiment is conducted to further explore the relationship of θ_c on the operation of the technique.

4.5.4.1. Meaning of Parameters

Using a random dataset, only three of four models were detected if $\alpha = 0.1$; whereas, all four were detected if $\alpha = 0.05$. This is shown in Listing 4-3. Based on the

process described in Section 4.3.2, and recalling that the overall technique is iterative, it becomes apparent that the fourth model was not prevalent enough to cause detection, even when data associated with the other data had been removed. This condition is expressed in Equation 4.14.

$$\frac{\Theta(D^w|C_i)}{\Theta(D^w)} < \alpha \quad (4.14)$$

Equivalently, it can be stated that the contamination detection technique will uncover a contamination model, C_i , if,

$$\frac{\Theta(D^w|C_i)}{\Theta(D^w)} > \alpha \quad (4.15)$$

This means that the final, “cleaned” data should have no regions of contamination greater than α -percent of the data.

The merging overlap parameter, γ , specifies the minimum correlation between two models, at a given iteration, that must be present for the models to merge into a single model. The ten correlations presented in the coverage matrix of Listing 1 suggest a polarity among real observations. Most models are either strongly ($\psi_{i,j} > 0.9$) or weakly ($\psi_{i,j} < 0.1$) correlated. The NNTP trace (Listing 4-2) confirms this generalization, albeit rather uninterestingly, as $\psi_{i,j}|i \neq j = 0$, suggesting a lack of correlation among all of the detected models.

The final parameter, θ_c , is the most difficult to define rigorously. It represents the severity of the contamination; or the range of the contamination relative to the range of the dataset. Therefore, one would expect a poorly chosen θ_c to affect the results of the contamination detection. Given a “tight” contamination region, choosing a low value of θ_c would over-detect, i.e., include data from outside the contamination region in the contamination model. This can be seen in Listing 4-4, which uses the same randomly generated data as Listing 4-3 and the values $\theta_c = 89.9$ and $\theta_c = 89.99$.

```

>> DetectedModels = process( D , 0.1 , 89.9 , 0.9 , MetaData , Debug );

Iteration 1

Hypothesized Models
[1]      (x1 == -3.000000)
[2]      (x2 == -3.000000)
[3]      (-1.000907 <= x3) & (x3 <= -0.994714)

Coverage Matrix
1.00000 0.59709 0.69141
0.68860 1.00000 0.68676
0.69222 0.59618 1.00000

Results of Model Merge
  1      2      3

Accepted Models
[1]      (x1 == -3.000000)
[2]      (x2 == -3.000000)
[3]      (-1.000907 <= x3) & (x3 <= -0.994714)

>> DetectedModels = process( D , 0.1 , 89.99 , 0.9 , MetaData , Debug );

Iteration 1

Hypothesized Models
[1]      (x1 == -3.000000)
[2]      (x2 == -3.000000)
[3]      (x3 == -1.000000)

Coverage Matrix
1.00000 0.59709 0.69098
0.68860 1.00000 0.68592
0.69229 0.59589 1.00000

Results of Model Merge
  1      2      3

Accepted Models
[1]      (x1 == -3.000000)
[2]      (x2 == -3.000000)
[3]      (x3 == -1.000000)

```

Listing 4-4 Effect of Critical Angle on Detection

The converse holds also; given a broad region of contamination selecting too large a value for θ_c can prohibit detection altogether. The next section will present experimental results designed to explore the effect of the critical angle on the detection process.

4.5.4.2. Effect of Critical Angle on Detection

The effect of the critical angle on contamination region detection was studied using a simulation experiment. The experimental trial involved generating 1000 points from a Gaussian mixture and running the single-variate contamination region detection process of Section 4.3.2 using this data. This trial was repeated 500 times per each value of the parameter set of the experiment. The level of contamination, equivalent to the

weight of the contaminating Gaussian component, was $w_C = 0.1$. Based on the discussion above, the detection level must be lower than this to detect the contamination. The value chosen was $\alpha = 0.05$. Since only a single contamination model exists, the effect of the merging parameter, γ , is moot. The remaining parameters of the experiment are described in Table 4-2.

Table 4-2 Simulation Experiment Parameters

Notation	Description	Qualitative Measure
μ_C/σ_N	Ratio of the mean of the contamination to the standard deviation of the “normal” data.	How close to the bulk of the “normal” data the contamination lies. Larger values indicate stronger contaminate isolation.
σ_C/σ_N	Ratio of the standard deviation of the contamination to the standard deviation of the “normal” data.	“Steepness” of the contamination relative to the “normal” data. Smaller values indicate a larger relative “steepness”.
θ_C	The critical angle of detection.	Minimum “steepness” required by contamination for detection.

The success of the contamination detection process was measured by the number of data classified incorrectly, either normal data modeled as contamination or contamination data not modeled, scaled by the overall size of the dataset. Equation 4.16 defines this metric, where C_{true} is the set of contamination data and $\hat{C}$ is the set of detected contamination data.

$$\varepsilon = \frac{\Theta(C_{true} \not\subset \hat{C}) + \Theta(\hat{C} \not\subset C_{true})}{\Theta(D)} \quad (4.16)$$

Figure 4-3 shows the error as a function of the ratio of standard deviations and the critical angle for a mean ratio of 2, e.g, $\varepsilon = f(\theta_C, \sigma_C/\sigma_N) | \mu_C/\sigma_N = 2$. The figure shows three regions of interest. For low values of the critical angle, $\theta_C \lesssim 75^\circ$, the error surface is dominated by a steep plane. In this region the error increases as the critical angle increases and the error is larger than the weight of the contamination. In this region points in the “normal” data are modeled as contamination because the constraint of “steepness” represented by the critical angle is overly relaxed.

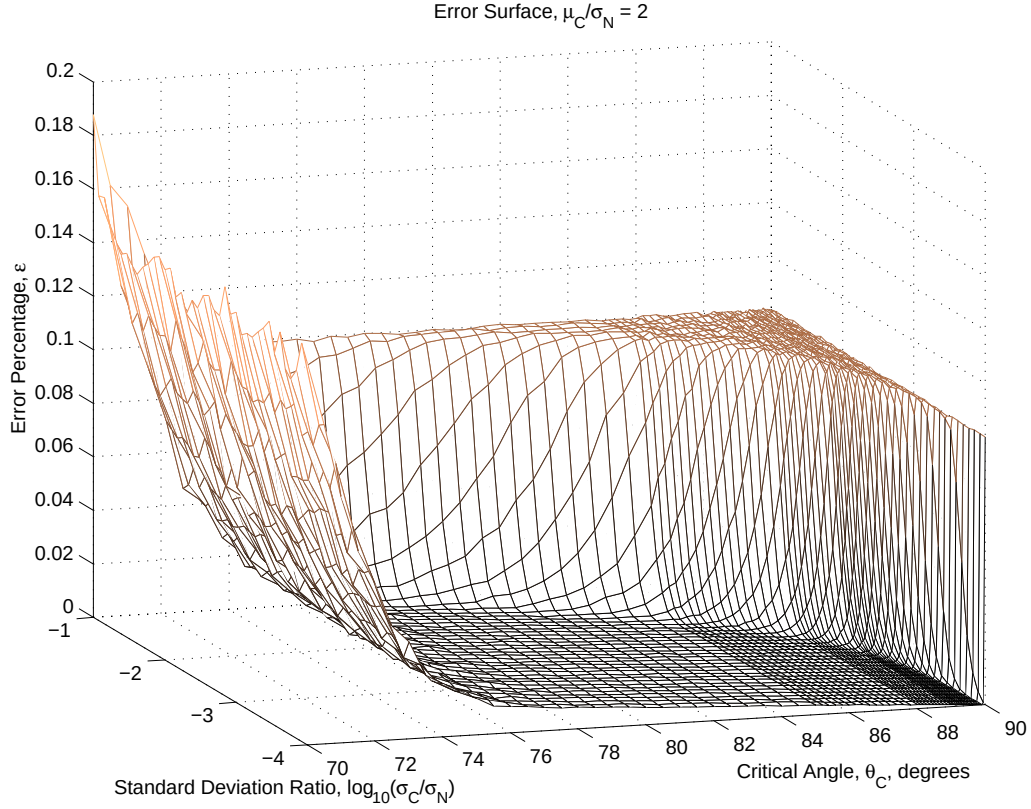


Figure 4-3 Error surface, mean ratio = 2

The second region of Figure 4-3 is a plane at $\varepsilon = 0.1$, which results from the critical angle being too stringent of a constraint to allow detection of the contamination model. This can be verified by noticing that, if $\hat{C} = \emptyset$, then $\varepsilon = w_C$. The third region of the error surface presented in Figure 4-3 is relatively low error region that is roughly planar. This region represents a near-optimal selection of parameters since the contamination is detected with a minimal amount of misclassified normal data.

Several trends exist in this region. For a given critical angle, the error is roughly constant until a threshold standard deviation ratio is reached, then the $\varepsilon = w_C$ region dominates. For a given standard deviation ratio, the error decreases as the critical angle increases; until a threshold is met, thereafter, the $\varepsilon = w_C$ region dominates. This trend is the manifestation of increasing the severity constraint resulting in less normal data being lumped with the contamination. Moreover, this trend implies an optimal critical angle, as a function of the standard deviation ratio, defined by Equation 4.17.

$$\theta_c^{opt}(\sigma_c/\sigma_N) \equiv \min_{\theta_c} [\varepsilon(\theta_c, \sigma_c/\sigma_N)] \quad (4.17)$$

Equation 4.17 does rely on a particular value of the mean ratio, μ_c/σ_N ; however, this has been omitted from the notation for simplicity.

Figure 4-4 shows the optimal critical angle as a function of the standard deviation ratio, for $\mu_c/\sigma_N = 2$. The figure shows an inverse relationship between the optimal critical angle and the standard deviation ratio. Equivalently, as the contamination becomes relatively “wider”, the “severity” constraint must be loosened to ensure detection. The optimal critical angle curve of Figure 4-4 represents the point where the “severity” constraint is just loose enough to allow detection; if it were relaxed further, more elements of the “normal” data would be associated with the contamination, and the error would be sub-optimal.

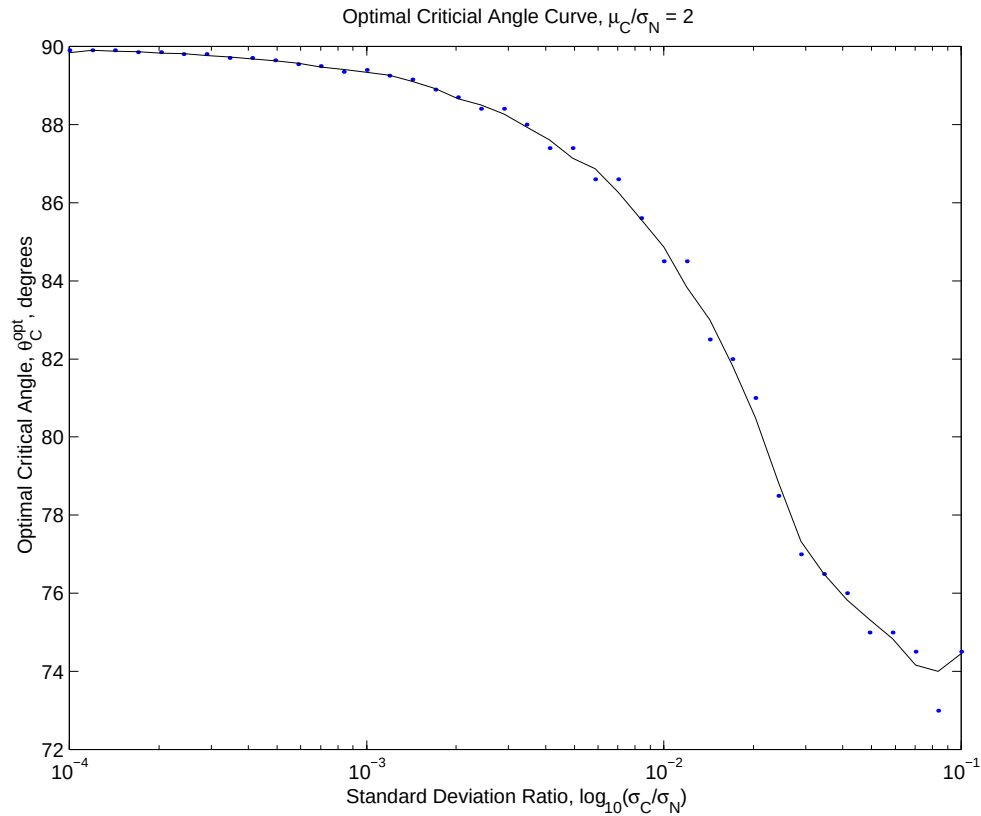


Figure 4-4 Optimal critical angle, mean ratio=2

The error surface is also a function of the mean ratio, μ_C/σ_N . Figure 4-5 shows the error surface for the case where $\mu_C/\sigma_N = 4$. The larger mean ratio implies that the contamination is located farther in the tails of the distribution of normal data; i.e., there is less normal data in the region of the contamination. Since little normal data exists in the region near the contamination, as the severity constraint is relaxed the error introduced from including normal data is small. This causes the region of near-optimal performance to be lower and flatter than the analogous region of Figure 4-3. However, a boundary does still exist between the near-optimal region and the region defined by $\varepsilon = w_C$, with the same implication as the prior case: given a standard deviation ratio, there exists a maximum critical angle beyond which detection is prohibited.

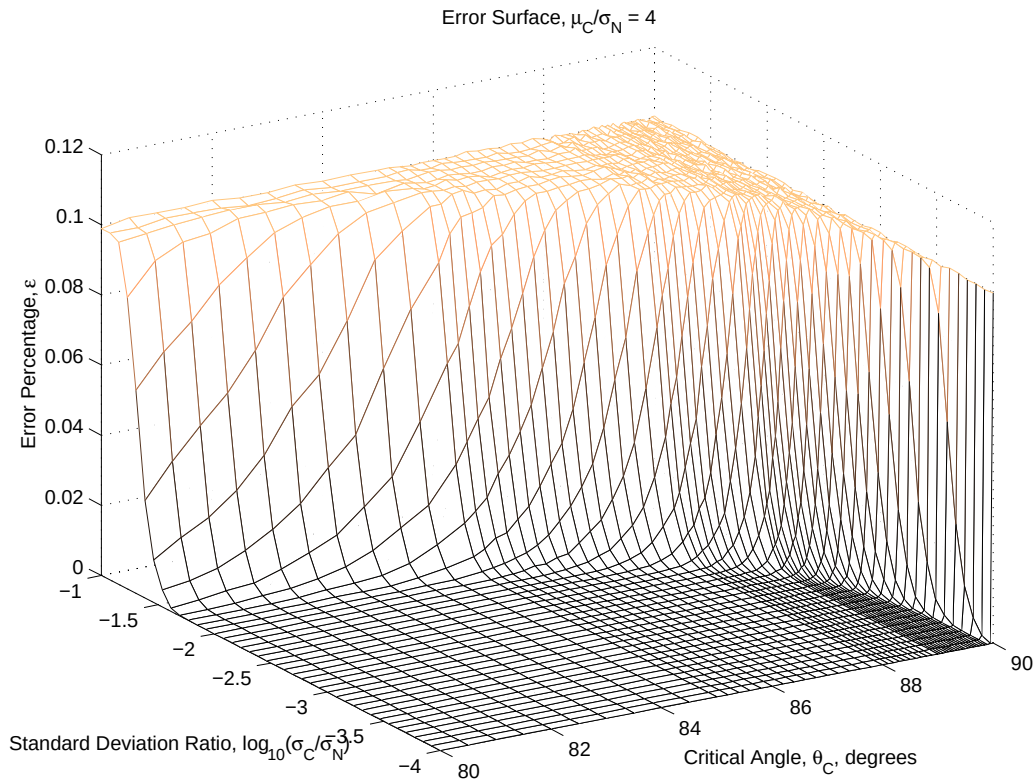


Figure 4-5 Error surface, mean ratio = 4

Comparing the error rates of the near-optimal regions of Figure 4-3 and 4-5 as a function of critical angle underscores two trends. First, relaxing the severity constraint will decrease performance, as more normal data is lumped with the contamination model.

Second, given a critical angle, the error is lower if the contamination is located farther in the tails of the distribution of the normal data. These points are illustrated by Figure 4-6, which shows the average error as a function of the critical angle, in the near-optimal regions of each error surface.

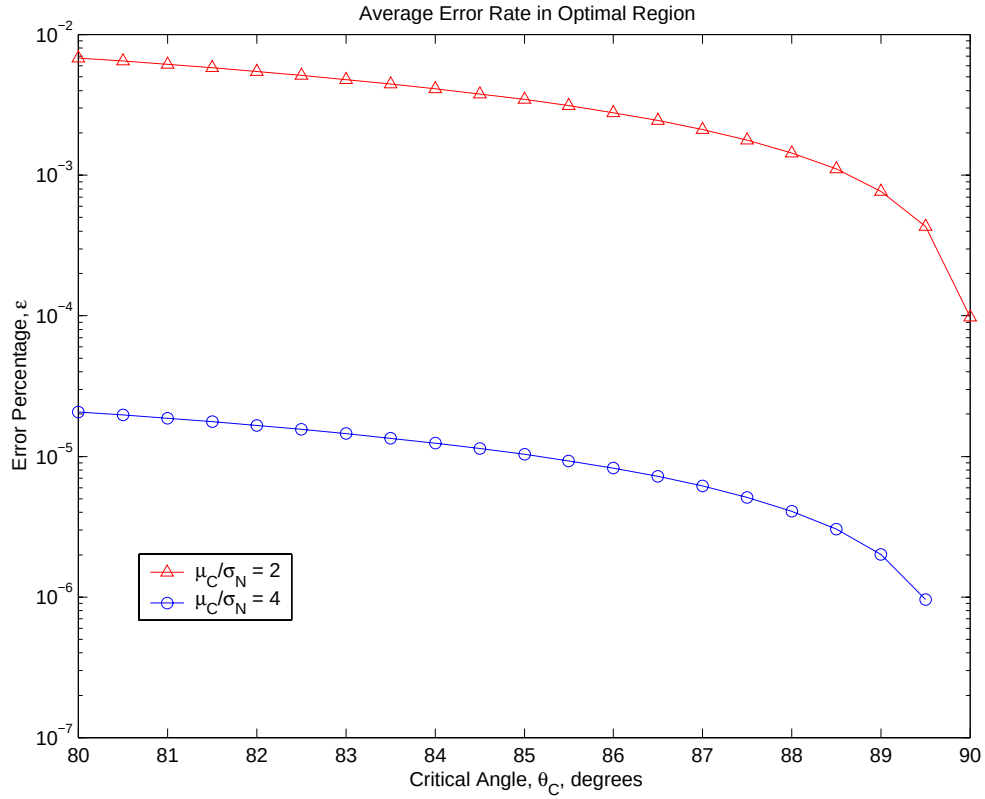


Figure 4-6 Average Error rate in near-optimal region

This section has examined the effects of the parameters, α , θ_c , and γ , on the performance of the contamination detection algorithm. The conclusions are summarized below.

- α represents the maximum amount of contamination that will be tolerated in the final dataset.
- θ_c represents the degree of severity or clustering anticipated of the contamination. Selection of θ_c involves a compromise between a sufficiently relaxed constraint, to detect potential widely spread

contamination, and an overly relaxed constraint, that will erroneously associate non-contaminated data with contaminated data.

- γ represents the minimum correlation between two detected models required to assume that those models are manifestations of the same underlying contamination model.

4.5.5. Scalability

The scalability of the algorithm was assessed using simulation experiments. Given a number of variables, N_v ; a number of contamination models, N_m ; and a dataset size, m , a random dataset was generated. This dataset had N_m contamination models, each with a randomly assigned probability of occurrence, equally likelihood of manifestation in each of the N_v variables, and an integer value taken from a uniform distribution in the range $\{-4,4\}$. The contamination detection algorithm was then performed and the number of floating point operations was measured, as a metric of scalability.

The first set of experiments varied the number of variables, N_v , and held the other two parameters constant at $N_m = 3$ and $m = 5000$. Figure 4-7 shows the result—as the number of variables in the dataset increases, the number of floating point operations (flops) increases linearly. This result is expected given that the contamination region detection process is repeated for each variable.

The next variable studied for scalability was the number of data points, m . The number of slope calculations done in the sliding-window contaminate detection process are proportional to the number of data; therefore, this variable is expected to have a linear effect on the number of floating point operations. These expectations were confirmed via experimentation. The experiment varied the size of the dataset and held the other two parameters fixed at $N_v = 5$ and $N_m = 4$. The number of floating point operations increases linearly as the size of the dataset is increased, as shown in Figure 4-8.

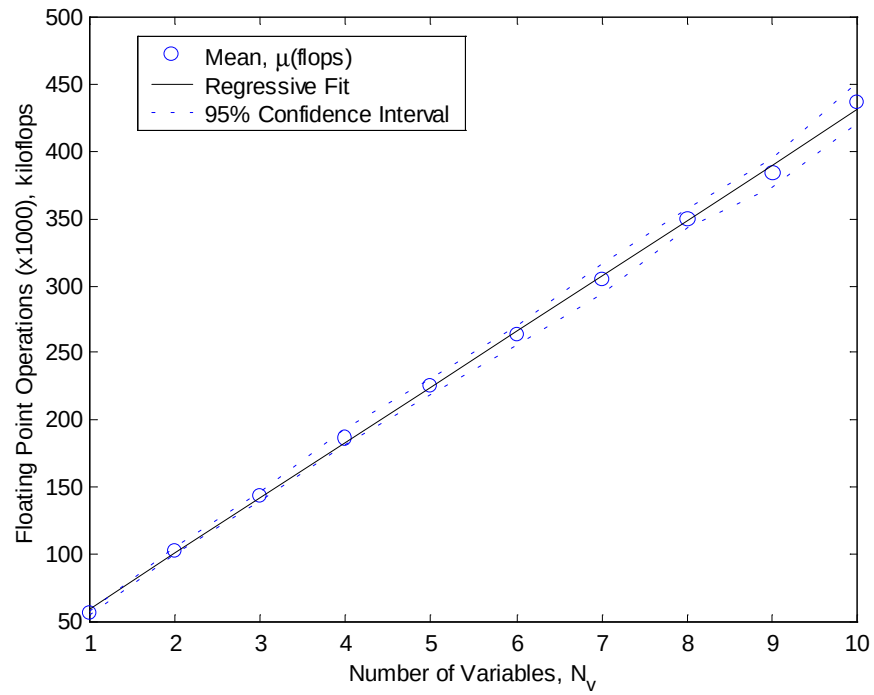


Figure 4-7 Effect of number of variables on scalability

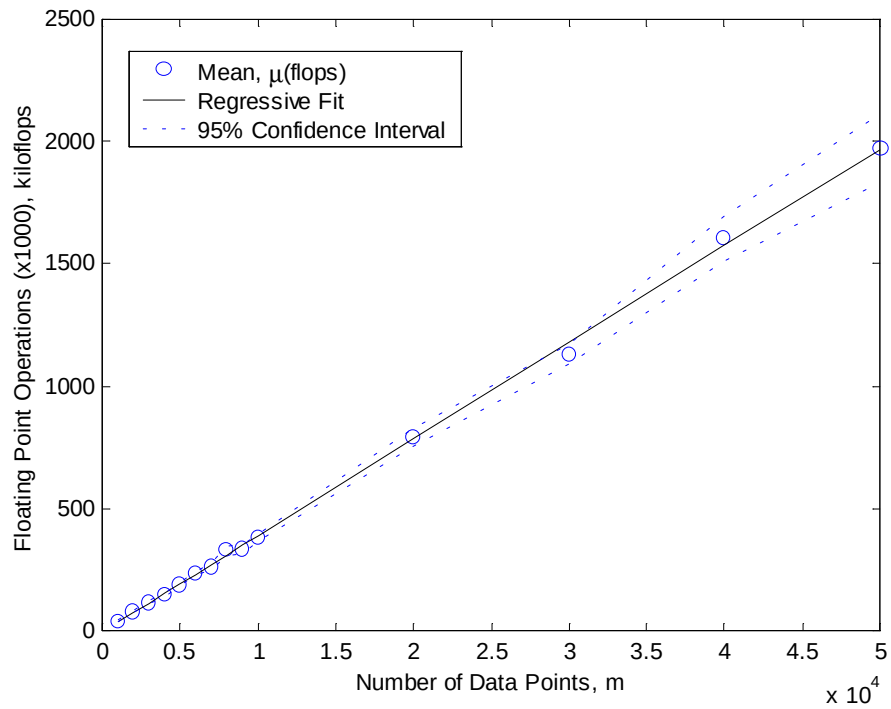


Figure 4-8 Effect of dataset size on scalability

The final experiment held constant $N_v = 5$ and $m = 5000$, varying the number of contamination models. The detection parameter α was chosen to be 90% of the median value of the contaminate probabilities. Figure 4-9 shows that the number of floating point operations increase rapidly, until about 5 models are present², thereafter, the number of operations tapers off. Figure 4-10 shows the average number of iterations required by the algorithm to converge. The number of iterations rises until about 5 models; thereafter, the number of iterations decreases. In the first region (increasing operations and iterations) the algorithm is effected by the selection of α . Some models are not detected on the first pass; therefore more iterations and more floating point operations are required to detect the contamination.

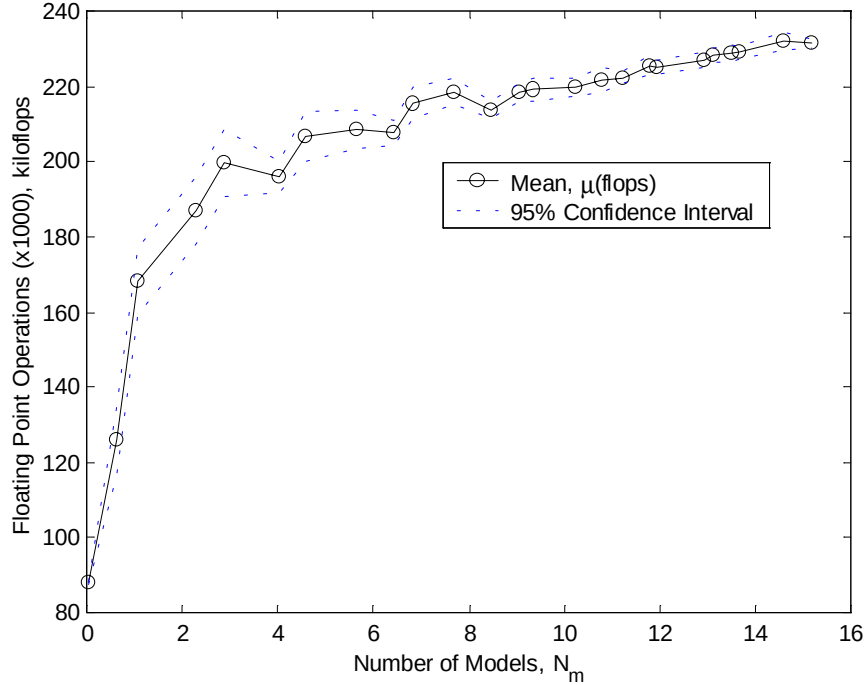


Figure 4-9 Effect of number of models on flops

² The figures plot the number of models detected, rather than the number of models in the random data. This accounts for the fact that the random data generator can output overlapping models, e.g., given a request for two models, nothing prevents the data generator from randomly picking “ $x_1=2$ ” twice.

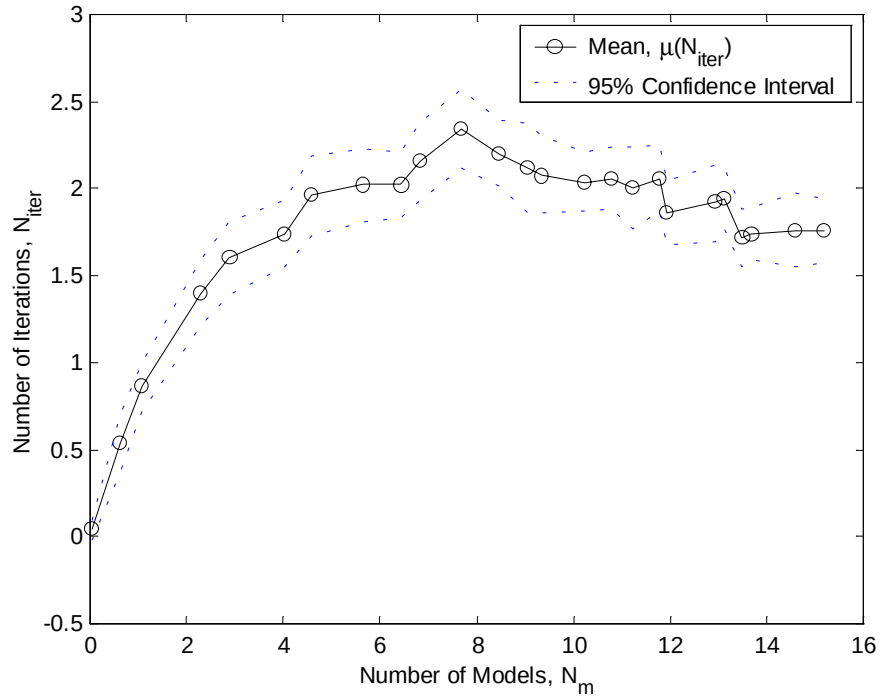


Figure 4-10 Effect of number of models on iterations

The second region is dominated by the effect of increasing the number of models. While the average number of iterations slowly decreases, the number of floating point operations increases. This represents the increased computation burden of removing more model-related data during the generation of the working data. However, this increased computation does not grow as rapidly as in the case of increasing the number of variables, or the case of increasing the number of data points. Therefore, the algorithm is more scalable with respect to the number of models.

4.6. Conclusions

This chapter began by formulating a problem, to develop a *parameterized, automated technique to detect and formalize contamination models using only a set of observations and metadata describing those observations*. A technique was developed to address this problem. This technique consisted of several sub-processes. Each of these sub-processes, listed below, were described:

- Censoring data based on existing contamination models;
- Single-variate contamination region detection;

- Contamination model formalization; and
- Contamination model merging.

The implementation of this technique was briefly discussed. Sample runs using real and simulated data showed the correctness of the method. The method was shown to be stable and was shown to provide a near minimal set of contamination models, hindered by increasing the data decontamination. The effects of the parameters of the algorithm on performance were studied using real and simulated data. Guidelines for the selection of these parameters were devised. The technique was found to be scalable, $O(n)$, as the number of datum or the number of independent variables within the data grow. Finally, the technique was found to be highly scaleable, $< O(n)$, as the number of contamination models increase.

Chapter 5: Estimation of Multimodal Distributions

Chapter 3 explored the characteristics of atypical WAN traffic and how it can interfere with modeling efforts. Chapter 4 presented a technique designed to account for one form of contamination: high clustered single, or nearly single-valued, contamination. However, this is not the only form of atypical traffic that can contaminate observations. Section 3.2.3 discussed the problem of site-specific network configuration, which can also lead to atypical traffic.

To account for the effect of site-specific network configurations, and more generally, to deal with multimodal network traffic (discussed in Section 2.2), a method for estimating the parameters of multimodal probability distributions is required. This chapter examines the problem of multimodal estimation, beginning with an examination of the site-specific atypical traffic context. Then two conventional estimation techniques, the method of moments and maximum likelihood estimation, are examined from the perspective of estimating the parameters of a two-component Gaussian mixture, which is used as a model of a multimodal distribution. A new technique for multimodal estimation, based on an extension of the midpoint of the shortest half and the length of the shorth presented in Chapter 2, is developed and implemented. This new technique is called the midpoint-distance profile method. Finally, these three techniques of multimodal estimation are studied to evaluate their statistical properties, numerical complexity, and extensibility.

5.1. Atypical Traffic and Multimodal Estimation

The effect of co-located NNTP servers on connection duration is depicted in Figure 5-1. The figure shows that observed non-error NNTP connections have a multimodal distribution of durations. When connections to the co-located server are removed, the distribution of durations becomes single-modal and favors the second mode.

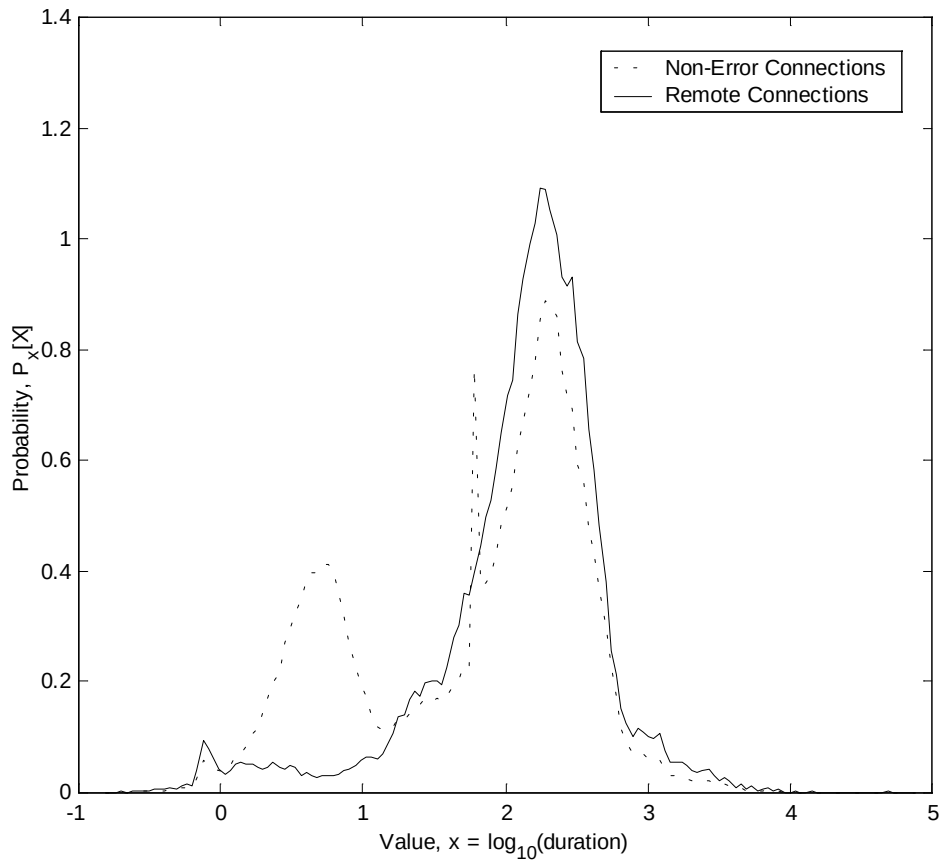


Figure 5-1 NNTP Duration PDF

The effect of local, site-specific traffic is not contamination of the sort studied in Chapter 4. Rather than being strongly clustered around a singular value, this contamination manifests itself as another mode present in a multimodal distribution. The technique of Chapter 4 locates strongly clustered contaminants; however, the co-location problem requires a technique to estimate the parameters of multimodal distributions. This chapter presents and studies techniques for this problem.

5.2. Conventional Estimation Techniques

Chapter 2 presented a general description of two common estimation methods: maximum likelihood estimation and the method of moments. Additionally, Chapter 2 overviewed the Gaussian mixture distribution. This section revisits these three topics, applying them to the specific case of a two-component Gaussian mixture.

5.2.1. Two-Component Gaussian Mixture

The probability distribution of a Gaussian mixture is given by Equation 2.1, repeated below.

$$\begin{aligned} f_Z(z) &= \sum_{i=1}^n w_i \phi(z, \mu_i, \sigma_i) \\ &= \sum_{i=1}^n \frac{w_i}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(z - \mu_i)^2}{2\sigma_i^2}\right) \end{aligned}$$

The weights of the components of a Gaussian mixture must sum to one; therefore, the probability distribution function of a two-component Gaussian mixture is given by:

$$f_Z(z) = \frac{w_1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(z - \mu_1)^2}{2\sigma_1^2}\right) + \frac{(1 - w_1)}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{(z - \mu_2)^2}{2\sigma_2^2}\right) \quad (5.1)$$

Examination of Equation 5.1 shows that the two-component Gaussian mixture has 5 independent parameters: $w_1, \mu_1, \sigma_1, \mu_2$, and σ_2 . Thus, given a data set hypothesized to be from a Gaussian mixture, a method for obtaining estimates of the five parameters is required. The method of moments and maximum likelihood solutions to this problem are discussed below.

5.2.2. Method of Moments

Section 2.3.3 outlines the method of moments in general. Given a distribution with N parameters, the first N analytic moments of the distribution function are calculated. By equating these analytic expressions with the values estimated from the data, a system of N equations is obtained. Estimates of the parameters can be obtained from this system of equations.

Equation 5.2 gives the first five analytic moments of the two-component Gaussian mixture. [15]

$$\begin{aligned} M_1 &= w_1\mu_1 + (1 - w_1)\mu_2 \\ M_2 &= w_1(\mu_1^2 + \sigma_1^2) + (1 - w_1)(\mu_2^2 + \sigma_2^2) \\ M_3 &= w_1(\mu_1^3 + 3\mu_1\sigma_1^2) + (1 - w_1)(\mu_2^3 + 3\mu_2\sigma_2^2) \\ M_4 &= w_1(\mu_1^4 + 6\mu_1\sigma_1^2 + 3\sigma_1^4) + (1 - w_1)(\mu_2^4 + 6\mu_2\sigma_2^2 + 3\sigma_2^4) \\ M_5 &= w_1(\mu_1^5 + 10\mu_1^3\sigma_1^2 + 15\mu_1\sigma_1^4) + (1 - w_1)(\mu_2^5 + 10\mu_2^3\sigma_2^2 + 15\mu_2\sigma_2^4) \end{aligned} \quad (5.2)$$

Estimates of the moments can be obtained from the data using Equation 2.14, repeated below.

$$\hat{M}_n = \frac{1}{m} \sum_{i=1}^m z_i^n$$

The system represented by Equation 5.2 is highly nonlinear; therefore, solving the system is non-trivial. This problem has been previously studied, and a synopsis of a technique for solving this system follows, based on the discussion presented by Johnson and Kotz. [15]

A ninth degree polynomial can be derived for $\phi = (\mu_1 - M_1)(\mu_2 - M_1)$. This polynomial exploits the central moments, M'_n , and is given by Equation 5.3.

$$\begin{aligned} 0 &= \sum_{i=1}^9 a_i \phi^i \\ a_0 &= -24(M'_3)^6 \\ a_1 &= -96(M'_3)^4 \kappa_4 \\ a_2 &= -63(M'_3)^2 \kappa_4^2 - 72(M'_3)^3 \kappa_5 \\ a_3 &= 288(M'_3)^4 - 108(M'_3) \kappa_4 \kappa_5 + 27 \kappa_4^2 \\ a_4 &= 444(M'_3)^2 \kappa_4 - 18 \kappa_5^2 \\ a_5 &= 90 \kappa_4^2 + 72(M'_3) \kappa_5 \\ a_6 &= 36(M'_3)^2 \\ a_7 &= 84 \kappa_4 \\ a_8 &= 0 \\ a_9 &= 24 \\ \kappa_4 &= M'_4 - 3(M'_2)^2 \\ \kappa_5 &= M'_5 - 10M'_2 M'_3 \end{aligned} \tag{5.3}$$

Using the estimated central moments, $\hat{M}'_n$, in place of the analytic ones, the roots of Equation 5.3 can be found numerically. Since M_1 lies between μ_1 and μ_2 , it follows that $\phi < 0$. Therefore, using only the negative roots of Equation 5.3, a set of plausible roots can be found, denoted as $\mathfrak{R}_\phi$.

Using the value $\phi' = \mu_1 + \mu_2 - 2M_1$, the relationship of Equation 5.4 holds.

$$2\phi^2 \phi'^2 + 4\phi M'_3 \phi' - 6\phi^3 - 3\kappa_4 \phi - (M'_3)^2 = 0 \tag{5.4}$$

Using a candidate value of ϕ from $\mathfrak{R}_\phi$ and the analytic moments, Equation 5.4 can be solved for ϕ' .

Equation 5.4 will yield two candidates for ϕ' ; but, these two solutions represent only one distinct set of parameters for a Gaussian mixture. Selection of the other root of Equation 5.4 will interchange the Gaussian components, e.g., the values of $\{w_1, \mu_1, \sigma_1\}$ using one root will be the values of $\{w_2, \mu_2, \sigma_2\}$ if the other root is used. Therefore, given plausible values for $\phi = (\mu_1 - M_1)(\mu_2 - M_1)$ and $\phi' = \mu_1 + \mu_2 - 2M_1$, the parameters of the distribution can be estimated, with one set of estimates for each value in $\mathfrak{R}_\phi$.

Using the definitions of ϕ and ϕ' , the polynomial of Equation 5.5 can be derived.

$$(\mu_{\{1,2\}} - M_1)^2 - \phi'(\mu_{\{1,2\}} - M_1) + \phi = 0 \quad (5.5)$$

The notation $\mu_{\{1,2\}}$ is meant to indicate the existence of a root corresponding to each of the two component means. By numerically finding the roots of Equation 5.5 and accounting for the offset of M_1 , the estimates of the component means, denoted $\hat{\mu}_1$ and $\hat{\mu}_2$, are obtained. The weights can be estimated using the expression for the first analytic moment (Equation 5.1), as shown below:

$$\begin{aligned} \hat{w}_1 &= \frac{\hat{M}_1 - \hat{\mu}_2}{\hat{\mu}_1 - \hat{\mu}_2} \\ \hat{w}_2 &= 1 - \hat{w}_1 \end{aligned} \quad (5.6)$$

Given estimates for the weights and the means of the components, the analytic expressions for the second and third moments (Equation 5.1) amount to a system of linear equations in the variables σ_1^2 and σ_2^2 . Estimates can be found using the estimated values of the other parameters and the moments, as given below:

$$\begin{bmatrix} \hat{\sigma}_1^2 \\ \hat{\sigma}_2^2 \end{bmatrix} = \begin{bmatrix} \hat{w}_1 & \hat{w}_2 \\ 3\hat{w}_1\hat{\mu}_1 & 3\hat{w}_2\hat{\mu}_2 \end{bmatrix}^{-1} \begin{bmatrix} \hat{M}_2 - \hat{w}_1\hat{\mu}_1^2 - \hat{w}_2\hat{\mu}_2^2 \\ \hat{M}_3 - \hat{w}_1\hat{\mu}_1^3 - \hat{w}_2\hat{\mu}_2^3 \end{bmatrix} \quad (5.7)$$

Thus, given a value of ϕ it is possible to obtain estimates for the parameters of the two-component Gaussian mixture, $\hat{\theta} = \{\hat{w}_1, \hat{\mu}_2, \hat{\sigma}_1, \hat{w}_2, \hat{\mu}_1, \hat{\sigma}_2\}$. Since more than one root to Equation 5.3 may exist, several heuristics exist to selecting the correct set of parameters. Any value of ϕ that leads to weight estimates that are less than zero or

greater than one can be disregarded. Similarly, any value of ϕ leading to negative values of $\hat{\sigma}_1^2$ or $\hat{\sigma}_2^2$ can be discounted. In the event that multiple solutions exist, the parameters that minimize the maximum relative discrepancy between the analytic and empirical moments are selected.

5.2.3. Maximum Likelihood Estimation

A general procedure for determining the maximum likelihood estimates of the parameters of a distribution is searching over the parameter space for the values that maximize the likelihood function, which is the probability density function assuming the parameters, i.e., $f_z(z|\bar{\theta})$. This maximization problem is transformed into a minimization problem using the negative log-likelihood function. Therefore, Equation 2.11, repeated below, expresses the maximum likelihood estimation problem.

$$\hat{\theta}_{MLE} = \arg \min_{\bar{\theta}} \left[- \sum_{i=1}^m \ln(f_z(z_i|\bar{\theta})) \right]$$

For notational simplicity, call the argument of the minimization $\rho(\bar{\theta}, \tilde{Z})$, where $\tilde{Z}$ is the observation set. The equation 2.11 becomes:

$$\begin{aligned} \hat{\theta}_{MLE} &= \arg \min_{\bar{\theta}} \rho(\bar{\theta}, \tilde{Z}) \\ \rho(\bar{\theta}, \tilde{Z}) &= \left[- \sum_{i=1}^m \ln(f_z(z_i|\bar{\theta})) \right] \end{aligned} \quad (5.8)$$

A two-component Gaussian mixture has six parameters, only five of which are independent. This parameter space is denoted by Equation 5.9.

$$\bar{\theta}_{GM} = \{w_1, \mu_1, \sigma_1, \mu_2, \sigma_2\} \quad (5.9)$$

Using the probability density function of a Gaussian mixture, the negative log-likelihood function is given by:

$$\begin{aligned} \rho_{GM}(w_1, \mu_1, \sigma_1, \mu_2, \sigma_2, \tilde{Z}) &= \\ &= - \sum_{i=1}^m \ln \left[\frac{w_1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(z_i - \mu_1)^2}{2\sigma_1^2}\right) + \frac{(1-w_1)}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(z_i - \mu_2)^2}{2\sigma_2^2}\right) \right] \end{aligned} \quad (5.10)$$

Using the Nelder-Mead simplex algorithm [39] as implemented in MATLAB, the values of the parameter space, $\hat{\theta}_{GM} = \{\hat{w}_1, \hat{\mu}_1, \hat{\sigma}_1, \hat{\mu}_2, \hat{\sigma}_2\}$, minimizing Equation 5.10 can be found. This approach is a numerical search technique that requires an initial guess, $\tilde{\theta}_{init}$. It is possible that more than one local minima of the objective function may exist; therefore, convergence can be sensitive to the initial guess.

5.3. Midpoint-Distance Profile Estimation

Using the two-component Gaussian mixture as a model for bimodal distributions, two standard estimation techniques, the method of moments and maximum likelihood, are used to derive estimation routines. This section extends the concepts of the midpoint of the shortest half and the location-free scale estimate presented in Section 2.5 to a generalized estimation technique for multimodal distributions. A brief review of the estimators from Section 2.5 is presented, followed by a definition of the midpoint distance profile. Using the midpoint-distance profile, methods for determining estimates of the locations, scales, and weights of Gaussian components are developed. Finally, the implementation of this technique is discussed.

5.3.1. Review of Location and Scale Estimation

Equation 2.23 defined the distance of the shortest half of a data set, $\tilde{Z}$, in terms of the order statistics. Equation 2.23 is repeated below.

$$d_{\min} = \min_{j \in \{1, \dots, m-v+1\}} d_j$$

$$d_j = z_{[j+v-1]} - z_{[j]}$$

$$v = \left\lfloor \frac{m}{2} \right\rfloor + 1$$

An estimate of the location is taken from the midpoint of the shortest half, as defined in Equation 2.24, repeated below.

$$\hat{\theta}_{MPSH} = \frac{1}{2} (z_{[k-v-1]} - z_{[k]})$$

$$k \therefore d_k = d_{\min}$$

Additionally, an estimate of the scale of a variable can be taken directly from the distance of the shortest half, as demonstrated by Equation 2.26, repeated below.

$$\hat{\theta}_{LFS} = c' d_{\min}$$

The constant c' depends on the underlying distribution of Z and is used to relate $\hat{\theta}_{LFS}$ to the particular scale parameter of that underlying distribution, e.g. σ for the Gaussian distribution.

These two estimators can be visualized with the aid of Figure 5-2. Assuming that Z is drawn from a unit normal distribution, the probability density of Z is depicted in Figure 5-2. After ordering the data, any given *half*, or a contiguous subset comprised of $v = \lfloor \frac{m}{2} \rfloor + 1$ elements, should correspond to a region under the probability density function having an area of 0.5. Figure 5-2 depicts two such regions. The first region, R_a , is the half beginning at $z = -1.75$ and the second half is the shortest half, which is centered on the mean, $\mu = 0$.

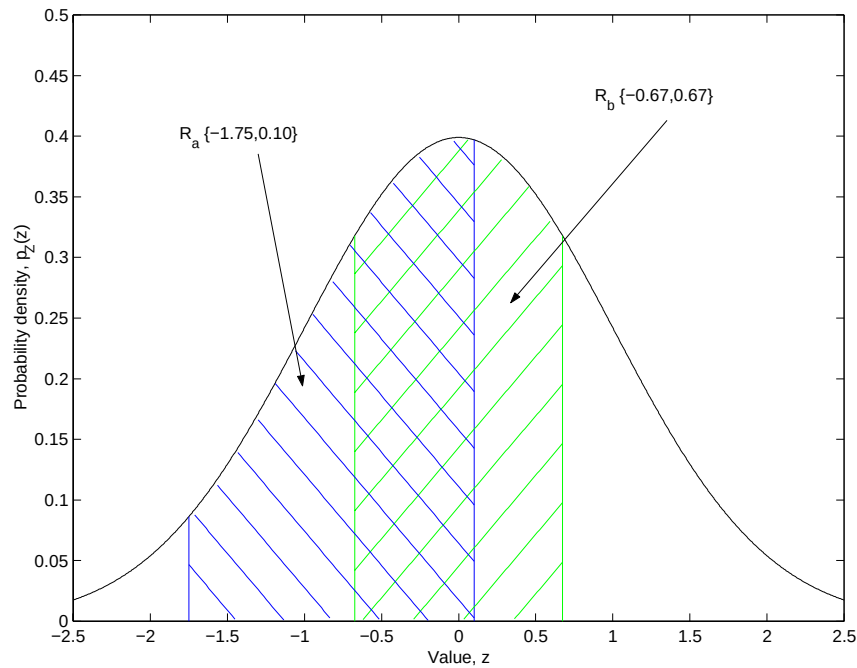


Figure 5-2 Comparison of two halves

Figure 5-2 illustrates the theory behind the midpoint of the shortest half. For a symmetric distribution function, the shortest half will be centered on the most frequent

value, or the mode. Using the midpoint of this region as an estimate of location, the mode can be estimated.

Similarly, the figure illustrates the theory behind the location-free scale estimate. Again, if the distribution is symmetric about the mode, the shortest half is centered about the mode. The lower bound of the region is given by the 0.25-percentile and the upper bound is given by the 0.75-percentile, as expressed in Equation 5.11.

$$\int_{-\infty}^{z_l} f_Z(z) dz = 0.25$$

$$\int_{z_u}^{\infty} f_Z(z) dz = 0.25$$
(5.11)

Therefore, the distance of the shortest half can be expressed in terms of the inverse distribution function, as in Equation 5.12.

$$d_{\min} = F_Z^{-1}(0.75) - F_Z^{-1}(0.25)$$
(5.12)

Equation 5.12 can be expressed in terms of the inverse distribution function for a unit variate, i.e. location parameter of zero and scale parameter of one, of the same underlying distribution, assuming a is the scale parameter of Z . Equation 5.13 shows this relationship.

$$d_{\min} = 2aF^{-1}(0.75)$$
(5.13)

5.3.2. Definition of the Midpoint-Distance Profile

The previous section reviewed estimators of location and scale that are based on the distances of *halves*, or contiguous regions of ordered data that are of size $v = \lfloor \frac{m}{2} \rfloor + 1$. These concepts can be generalized for an arbitrary fraction of the data. [18] This section describes this generalization and defines the midpoint distance profile. The properties of this profile are then examined using a Gaussian distribution and a two-component Gaussian mixture.

Generalizing the definition of v from a half to an arbitrary fraction of data, α , yields Equation 5.14.

$$v = \lfloor \alpha m \rfloor + 1$$
(5.14)

Equation 5.15 gives the distance and midpoint of a fraction of the data, which is an extension of Equations 2.23 and 2.24.

$$\left. \begin{aligned} d_j &= z_{[j+v-1]} - z_{[j]} \\ mp_j &= \frac{1}{2}(z_{[j]} + z_{[j+v-1]}) \end{aligned} \right\} j \in \{1, 2, \dots, m - v + 1\} \quad (5.15)$$

The midpoint-distance profile is obtained by plotting mp_j on the abscissa and d_j on the ordinate.

Given the previous discussion, the minimum of the midpoint-distance profile corresponds to an estimate of the location of the mode of a distribution. Generalizing Equation 5.13 in terms of α and assuming a Gaussian distribution, an estimate of the standard deviation can be obtained, as shown in Equation 5.16.

$$\hat{\sigma} = \frac{d_{\min}}{2P^{-1}\left(0.5 + \frac{\alpha}{2}\right)} = \frac{d_{\min}}{2\Phi^{-1}\left(\frac{1}{2}(1 + \alpha)\right)} \quad (5.16)$$

Figure 5-3 shows the midpoint-distance profile for a random sample of Gaussian variates using $\alpha = 0.5$ and $\alpha = 0.1$. For the data of Figure 5-3, $m = 10,000$, $\mu = 4$, and $\sigma = 0.5$. The figure illustrates that each profile has a minimum near the mode of the underlying distribution. The profile for $\alpha = 0.1$ is smaller-valued than for $\alpha = 0.5$, which is expected, as the distance spanned by a tenth of the data should be less than the distance spanned by half of the data. Additionally, the profile for $\alpha = 0.1$ is flatter in the region near the mode. This behavior is the result of taking a smaller window of data; near the mode, the distance will not be as sensitive to the location of the midpoint.

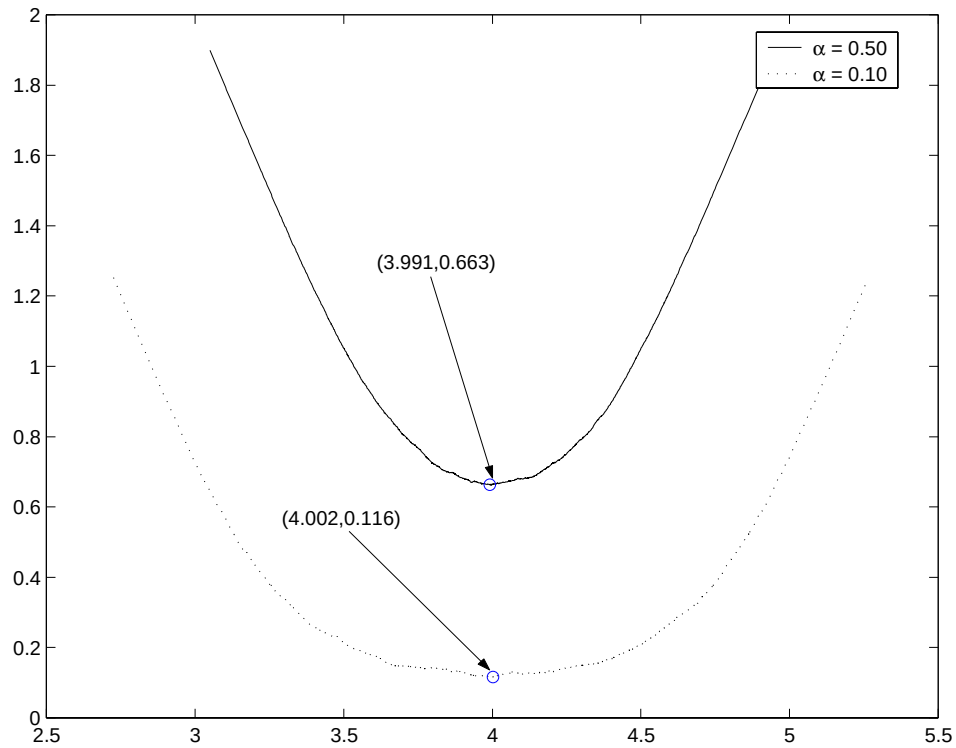


Figure 5-3 Comparison of two midpoint-distance profiles

Despite these differences, the two profiles yield comparable estimates for the parameters of the Gaussian distribution, as shown in Table 5-1. Estimates for the mode were taken directly from the midpoint of the shortest half, $\hat{\mu} = mp_k \ni d_k = d_{\min}$, and estimates of the standard deviation were obtained using Equation 5.16. The table illustrates that the midpoint-distance profile can be used to generate estimates for the parameters of a Gaussian distribution.

Table 5-1 Estimates from Two Profiles

Parameter	Actual Distribution	$\alpha = 0.5$ Estimate	$\alpha = 0.1$ Estimate
Mode, μ	4.0	3.991	4.002
Standard Deviation, σ	0.5	0.4916	0.4625
Shortest Distance, $d_{\min}$		0.6631	0.1162

Given the desire to estimate parameters of multimodal distribution, such as the Gaussian mixture, the profile of a two-component Gaussian mixture is examined. The

data set consists of 15,000 observations from a Gaussian mixture with $\{w_1 = 0.4, \mu_1 = -2, \sigma_1 = 1, w_2 = 0.6, \mu_2 = 3, \sigma_2 = 2.5\}$. Figure 5-4 shows the midpoint-distance profile for $\alpha = 0.1$.

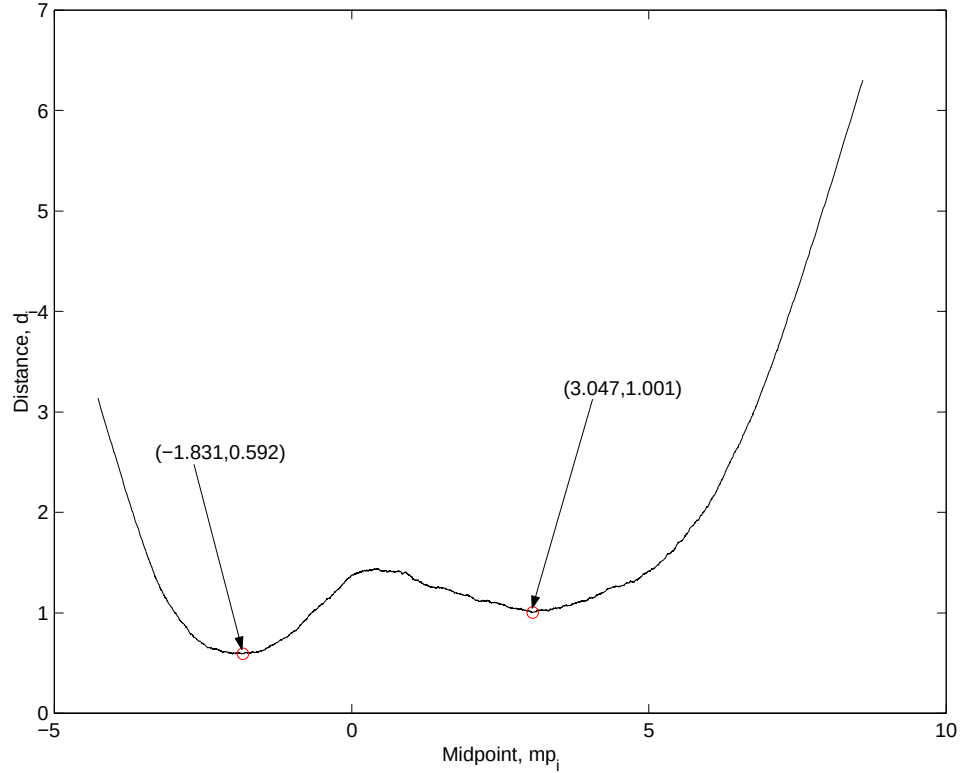


Figure 5-4 Midpoint distance profile for a bimodal distribution

The profile shows two local minima, corresponding roughly to the two means of the Gaussian mixture. However, the standard deviations and the weights are not immediately obvious from the profile. Methods for estimating the full set of parameters from the profile are discussed in the following three sections.

5.3.3. Estimation of Location

As alluded to above, estimates of the location of each component of a Gaussian mixture can be found from the location of local minima on the midpoint-distance profile. Equation 5.17 denotes these estimates.

$$\hat{\mu}_i = mp_j \ni \min_{j \in \mathfrak{N}_i} d_j \quad (5.17)$$

Thus, for the distribution depicted in Figure 5-4, the following estimates can be obtained:

$$\hat{\mu}_1 = -1.831 \text{ and } \hat{\mu}_2 = 3.047 .$$

The term $\mathfrak{R}_i$ denotes the region of the profile in the neighborhood of the i^{th} local minima. The method used to determine this region is discussed in Section 5.3.6; however, the following sections will make use of the notation as well.

5.3.4. Estimation of Scale

By extending the concept behind the length of the shorth, an estimate of scale can be developed. This section explores this estimate and its sensitivity to the weight of the Gaussian components. It is demonstrated that this estimate is highly sensitive to the weight of the component, so another estimate of scale is proposed, based on the relative distance between μ and $\mu + \sigma$.

5.3.4.1. Length of the Shorth

The length of the shorth, as discussed in Section 5.3.1, can be generalized to estimate the scale parameter of a Gaussian component. Assuming that in the region $\mathfrak{R}_i$ the midpoint-distance profile is dominated by the i^{th} component, the estimate of scale can be taken from Equation 5.16, with one modification. As Figure 5-5 shows, the minimum distance will increase as w_i decreases.

The trend is due to α being defined globally, i.e. for the entire distribution, but the minimum distance being a local property, i.e. only for a component. This can be expressed in terms of the component's contribution to the PDF, as in Equation 5.18.

$$\int_{-d_{\min}/2}^{d_{\min}/2} w_i \phi(z) dz = \alpha \quad (5.18)$$

$$\int_{-d_{\min}/2}^{d_{\min}/2} \phi(z) dz = \frac{\alpha}{w_i}$$

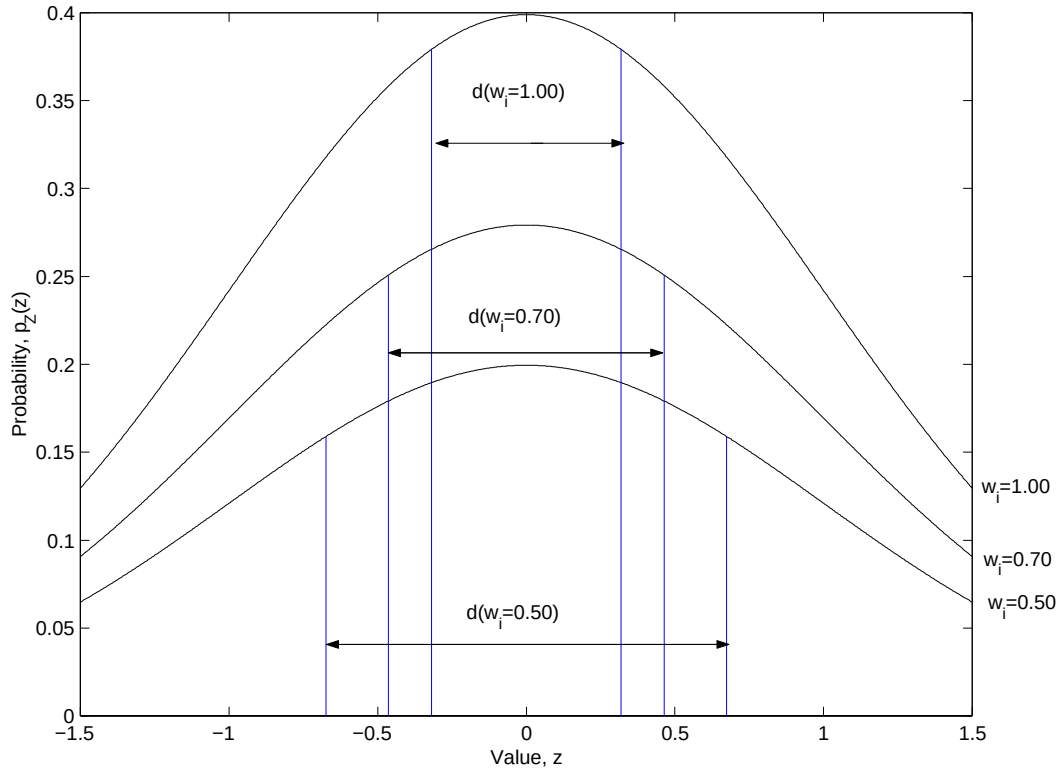


Figure 5-5 Minimum distances dependency on component weight

This is the same general relationship that governs the estimate obtained from the length of the shorth; however, the fraction of data, α , has been scaled to be α/w_i . This can be explained as follows. For a region dominated by a component of weight w_i to contribute an area of α to the global distribution function, the equivalent single component contribution is α/w_i . Thus, the estimate of scale is given by Equation 5.19.

$$\hat{\sigma}_i = \kappa \min_{j \in \mathcal{R}_i} d_j \quad (5.19)$$

$$\kappa(\alpha, w_i) = \left\{ 2\Phi^{-1} \left(\frac{1}{2} \left(1 + \frac{\alpha}{w_i} \right) \right) \right\}^{-1}$$

For an unbiased estimate of the standard deviation, $d_{\min}$ must be scaled by a factor that depends on the component's weight and the parameter α . This is problematic, because the component's weight is unknown and has to be estimated. Moreover, the

correction factor, κ , is highly sensitive to the component's weight, as shown in Figure 5-6.

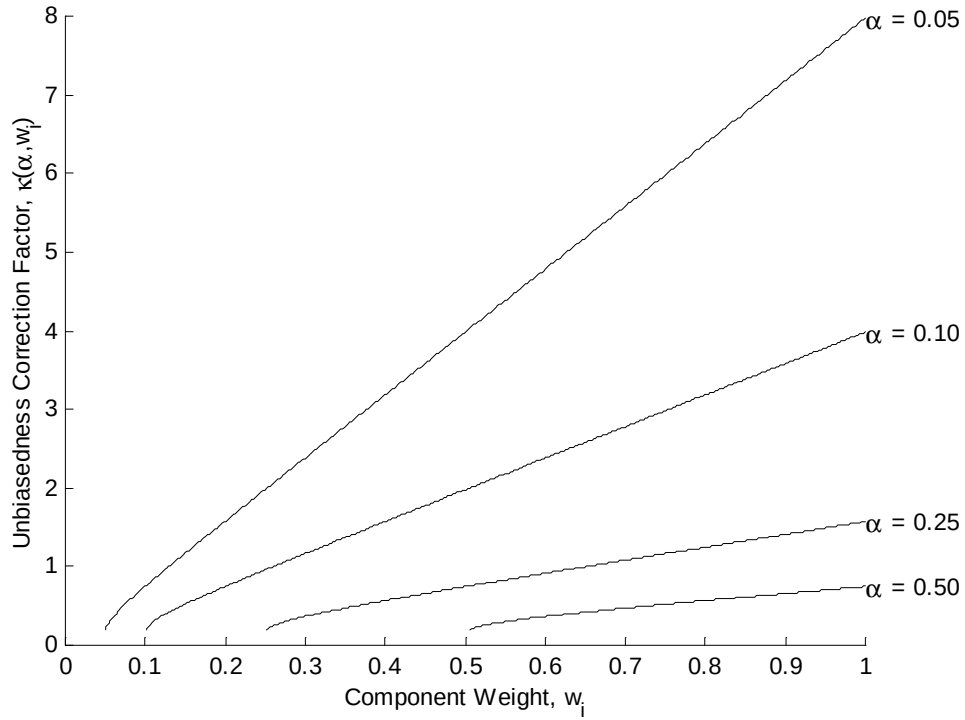


Figure 5-6 Sensitivity of correction factor to component weight

Without a priori knowledge of the weights, the scale estimates can occupy a very large range of plausible values. For the profile of Figure 5-3, the following range of estimates is obtained.

$$\min_{w_i} \kappa(0.1, w_i) = 0.1934 \quad \max_{w_i} \kappa(\alpha, w_i) = 3.979$$

$$\hat{\sigma}_1 \in \{0.1145, 2.356\}$$

$$\hat{\sigma}_2 \in \{0.1936, 3.983\}$$

However, if the knowledge of the weights is used, estimates are much closer to the correct values are obtained:

$$\hat{\sigma}_1 = \kappa(0.1, 0.4) 0.592 = 1.569 \cdot 0.592 = 0.929$$

$$\hat{\sigma}_2 = \kappa(0.1, 0.6) 1.001 = 2.376 \cdot 1.001 = 2.378$$

5.3.4.2. Scale Estimation using Relative Distances

Estimation based on a generalization of the length of the shorth is highly sensitive to the (unknown) value of the component weight; therefore, an alternative method of estimating the scale is proposed. This method is based on the relative distance, or length, of the α -fraction of data located around μ_i to the length of that centered around $\mu_i + \sigma_i$. This method is shown to be less sensitive to the value of the component weight.

Consider two regions of the Gaussian probability function. The first region has an area of α and a midpoint of the mean. The second region has the same area, but has a midpoint of $\mu + \sigma$. Figure 5-7 shows these regions for $w_i = 1$ and $w_i = 0.7$.

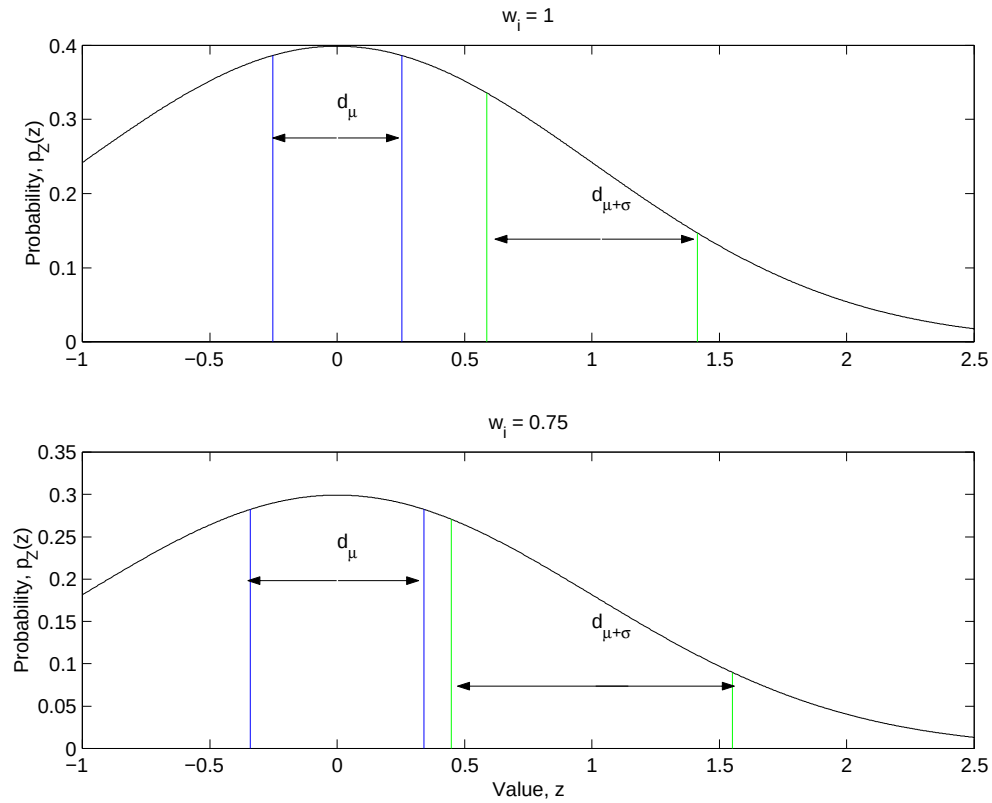


Figure 5-7 Relative distances

As with the length of the shorth, for a Gaussian component in the region where one mode dominates, the probability density function can be viewed in terms of an effective fraction of the data, $\frac{\alpha}{w_i}$. The distances are defined in Equation 5.20.

$$d_{\mu} = 2\sigma_i \Phi^{-1}\left(\frac{1}{2}\left(1 + \frac{\alpha}{w_i}\right)\right) \quad (5.20)$$

$$d_{\mu+\sigma} = b - a \quad (a, b) \ni \frac{a+b}{2} = \sigma \wedge \Phi(b) - \Phi(a) = \frac{\alpha}{w_i}$$

The ratio of $d_{\mu+\sigma}$ to d_{μ} is scale invariant (see Appendix D); therefore, it is only a function of $\frac{\alpha}{w_i}$. This allows a method for estimating the scale of a component using the local minima of the midpoint distance profile, $d_{\min,i} = \min_{j \in \mathfrak{R}_i} d_j$ and $mp_{\min,i}$. The distance can be multiplied by the ratio $\frac{d_{\mu+\sigma}}{d_{\mu}}$ to obtain the distance of the α -fraction of data with a midpoint of σ_i . The point on the profile nearest the $d_{\min,i}$ having this distance can be found. The corresponding midpoint, $mp_{\mu+\sigma}$, can be used to estimate the standard deviation. Equation 5.21 summarizes this procedure.

$$\hat{\sigma}_i = \left| mp_{\min,i} - mp_{\mu+\sigma} \right|$$

$$mp_{\min,i} \ni d_{\min,i} = \min_{j \in \mathfrak{R}_i} d_j \quad (5.21)$$

$$mp_{\mu+\sigma} \ni d_{\mu+\sigma} = \frac{d_{\mu+\sigma}}{d_{\mu}} d_{\min,i}$$

As with the length of the shorth, accurate estimation of scale requires prior knowledge of the component's weight, since $\frac{d_{\mu+\sigma}}{d_{\mu}}$ is a function of α and w_i . Figure 5-7 shows that as w_i decreases both $d_{\mu+\sigma}$ and d_{μ} increase, implying that the ratio may be less sensitive to the component weight than the length of the shorth. Figure 5-8 confirms this conjecture, showing that, as a function of α and w_i , the distance ratio is much less variable than the correction factor depicted in Figure 5-6.

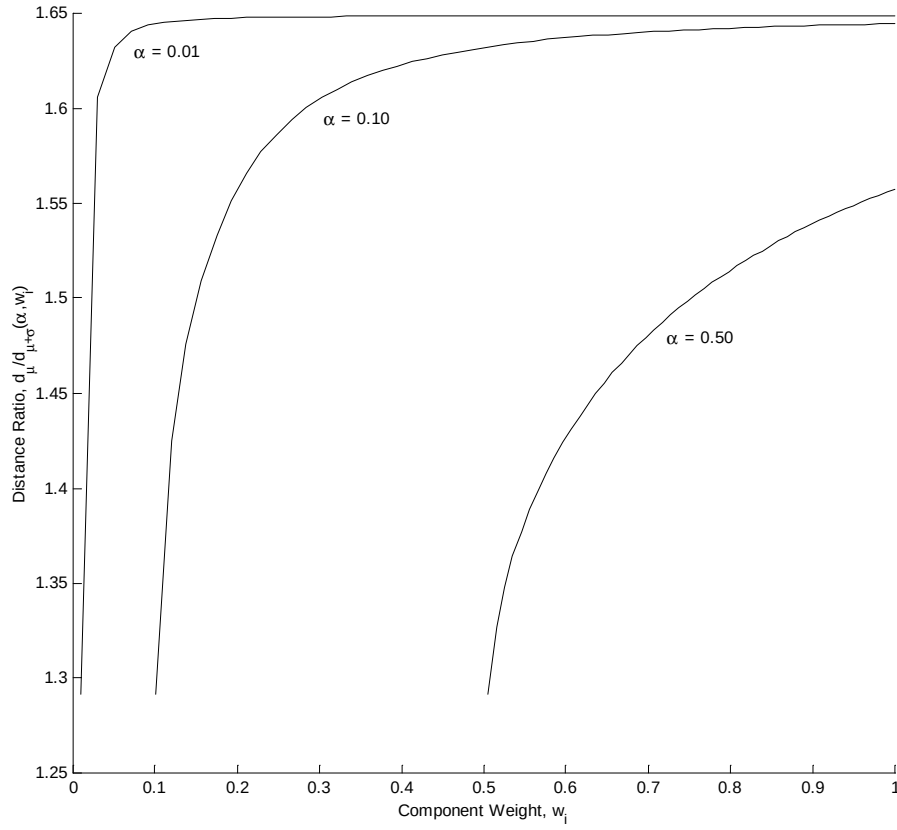


Figure 5-8 Effect of component weight on distance ratio

Without prior knowledge of the component weights, the scale estimates will occupy a range of values. However, this range is less than the range using the length of the shorth method. The ranges for the profile of Figure 5-4 are shown below.

$$\min_{\alpha=0.1} \frac{d_{\mu+\sigma}}{d_{\mu}} = 1.291 \quad \max_{\alpha=0.1} \frac{d_{\mu+\sigma}}{d_{\mu}} = 1.644$$

$$\hat{\sigma}_1 \rightarrow d_{\mu+\sigma} \in 0.592\{1.291, 1.644\} = \{0.7643, 0.9733\}$$

$$\hat{\sigma}_1 = |mp_{\mu+\sigma} - mp_{\min, i}| \in \{0.795, 1.110\}$$

$$\hat{\sigma}_2 \rightarrow d_{\mu+\sigma} \in 1.001\{1.291, 1.644\} = \{1.292, 1.646\}$$

$$\hat{\sigma}_2 = |mp_{\mu+\sigma} - mp_{\min, i}| \in \{1.562, 2.397\}$$

If knowledge of the weights is utilized, the estimates are much closer to the correct values and they are extremely close to the estimates obtained from the length of the short method. These estimates are shown below.

$$\begin{aligned}\hat{\sigma}_1 &\rightarrow \left. \frac{d_{\mu+\sigma}}{d_\mu} \right|_{\alpha=0.1, w_i=0.4} = 1.623 \\ d_{\mu+\sigma} &= 1.623 \cdot 0.592 \\ \hat{\sigma}_1 &= |mp_{\mu+\sigma} - mp_{\min,i}| = 1.098\end{aligned}$$

$$\begin{aligned}\hat{\sigma}_2 &\rightarrow \left. \frac{d_{\mu+\sigma}}{d_\mu} \right|_{\alpha=0.1, w_i=0.6} = 1.637 \\ d_{\mu+\sigma} &= 1.637 \cdot 1.001 \\ \hat{\sigma}_2 &= |mp_{\mu+\sigma} - mp_{\min,i}| = 2.387\end{aligned}$$

For this technique to be useful, a method for estimating the scale without prior knowledge of the weights is needed. This is achieved by using the average value of the distance ratio, denoted $\delta(\alpha)$ and defined in Equation 5.22.

$$\delta(\alpha) = \frac{1}{1-\alpha} \int_{\alpha}^1 \frac{d_{\mu+\sigma}}{d_\mu}(\alpha, w_i) dw_i \quad (5.22)$$

Equation 5.22 describes the average value of the distance ratio over the range of plausible values for the component weight, $w_i \in \{\alpha, 1\}$. Figure 5-8 shows the value of $\delta(\alpha)$ for several values of α . The figure also shows that the linear relation of Equation 5.23 can approximate the function.

$$\delta(\alpha) \approx 1.656 - 0.359\alpha \quad (5.23)$$

Using the linear approximation of the average value of the distance ratio, the algorithm of Equation 5.21 can be modified to provide a scale estimate without prior knowledge of the component weight. Equation 5.24 shows this estimation procedure.

$$\begin{aligned}\hat{\sigma}_i &= |mp_{\min,i} - mp_{\mu+\sigma}| \\ mp_{\min,i} &\ni d_{\min,i} = \min_{j \in \mathfrak{R}_i} d_j \\ mp_{\mu+\sigma} &\ni d_{\mu+\sigma} = (1.656 - 0.359\alpha) d_{\min,i}\end{aligned} \quad (5.24)$$

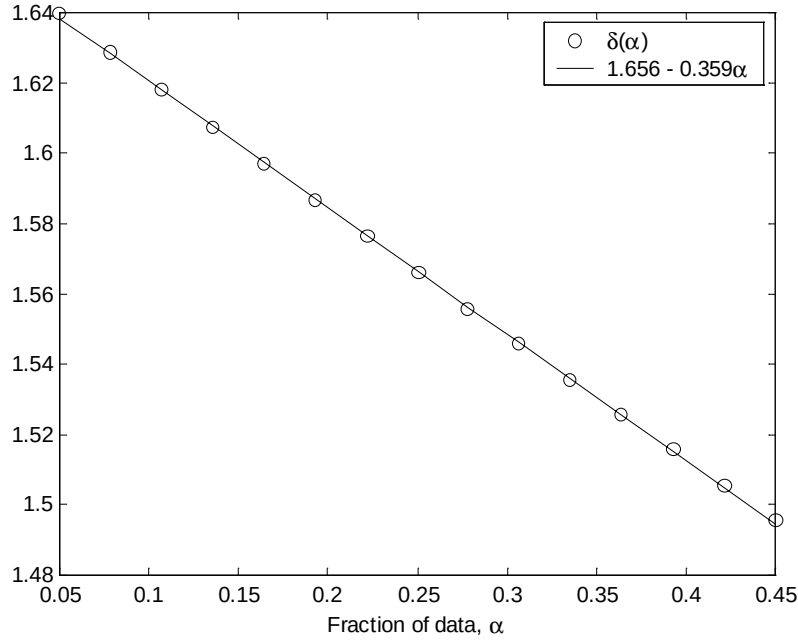


Figure 5-9 Average distance ratio and linear approximation

Using Equation 5.24, estimates can be obtained from the profile of Figure 5-4, as shown below.

$$\delta(0.1) \approx 1.656 - 0.359 \cdot 0.1 = 1.620$$

$$\hat{\sigma}_1 \rightarrow d_{\mu+\sigma} = 1.620 \cdot 0.592 = 0.959$$

$$\hat{\sigma}_1 = |mp_{\mu+\sigma} - mp_{\min,i}| = 1.09$$

$$\hat{\sigma}_2 \rightarrow d_{\mu+\sigma} = 1.620 \cdot 1.001 = 1.622$$

$$\hat{\sigma}_2 = |mp_{\mu+\sigma} - mp_{\min,i}| = 2.369$$

5.3.5. Estimating Component Weight

Using the estimates of scale obtained from the relative distances, estimates of component weight can be obtained from the length of the shorth. Equation 5.19 can be rearranged to obtain the estimates:

$$\hat{w}'_i = \alpha \left\{ 2\Phi\left(\frac{d_{\min,i}}{2\hat{\sigma}_i}\right) - 1 \right\}^{-1} \quad (5.25)$$

The estimates given by Equation 5.25 may not satisfy the constraint of totaling one, required for a valid Gaussian mixture. Therefore, the estimates are rescaled via Equation 5.26.

$$\hat{w}_i = \frac{\hat{w}'_i}{\sum_k \hat{w}'_k} \quad (5.26)$$

5.3.6. Summary

This section presented the theory behind the midpoint-distance profile technique for estimation of bimodal Gaussian mixtures. The estimates of the means of the components are given by finding the minimum value in the region of a local minima, as shown in Equation 5.17. Estimates of scale are obtained by the relative distance technique described in Section 5.3.4.2, as per Equation 5.24. Finally, Equations 5.25 and 5.26, derived from the length of the shorth, give estimates of the component's weight. The overall process is summarized in Equation 5.27.

$$\begin{aligned} \hat{\mu}_i &= mp_j \ni \min_{j \in \mathfrak{R}_i} d_j \\ \hat{\sigma}_i &= |\hat{\mu}_i - mp_{\mu+\sigma}| \\ mp_{\mu+\sigma} &\ni d_{\mu+\sigma} = (1.656 - 0.359\alpha) \min_{j \in \mathfrak{R}_i} d_j \\ \hat{w}_i &= \frac{\hat{w}'_i}{\sum_k \hat{w}'_k} \\ \hat{w}'_i &= \alpha \left\{ 2\Phi\left(\frac{d_{\min,i}}{2\hat{\sigma}_i}\right) - 1 \right\}^{-1} \end{aligned} \quad (5.27)$$

The next section explores several issues involved with practical implementation of the theoretical technique described in this section and the solutions to these issues.

5.4. Midpoint-Distance Profile Implementation

Estimation of the parameters of a Gaussian mixture using the midpoint-distance profile method requires several adaptations for implementation. The theory of Section 5.3 is complicated by irregularities, or “noise,” in the profile caused by the random nature of

the observations. This noise confounds efforts both to locate the region near a minimum, $\mathfrak{R}_i$, and to estimate the parameters from the profile. This section explores the problem of noisy profiles, and describes the adaptations made to the midpoint-distance profile technique to facilitate implementation.

5.4.1. Profile Noise

The midpoint-distance profile technique relies on the ability to locate local minima and to evaluate the inverse of the profile. Figure 5-10 shows the noise in the profile that complicates both of these efforts. More than one local minimum may exist in the region of a minimum of the distribution. Also, because of the variability of the profile, more than one value may exist for $d_{\mu+\sigma}$ and correspondingly for $mp_{\mu+\sigma}$. Consequently, multiple estimates for $\hat{\sigma}_i$ may occur.

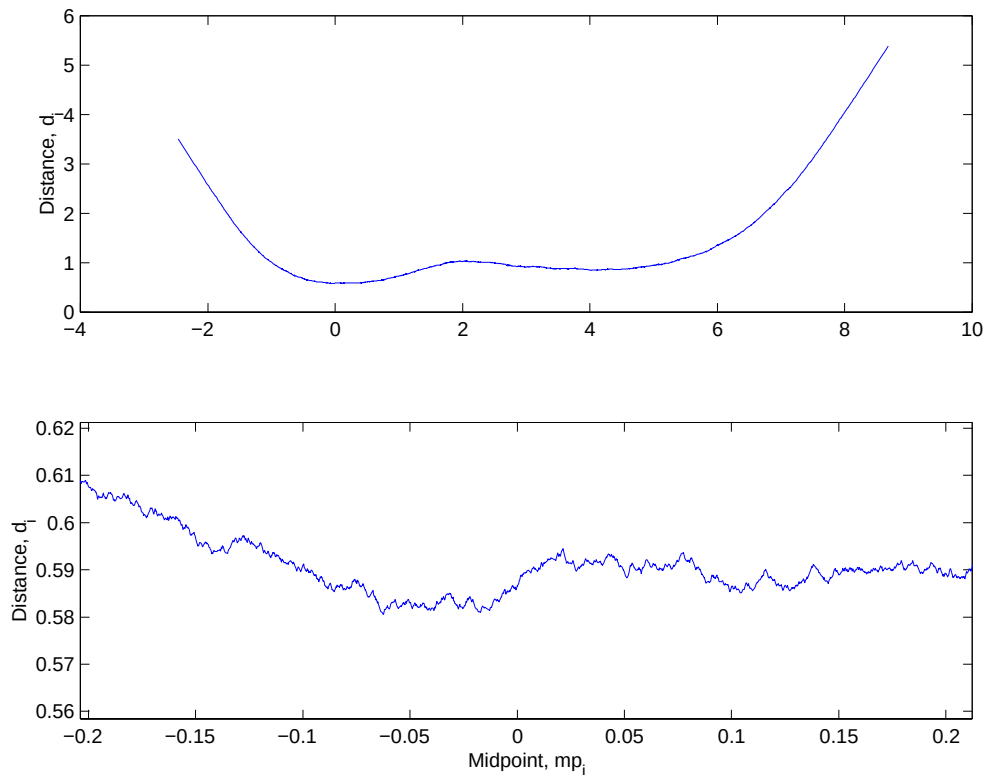


Figure 5-10 Problem of noise in a profile

Two approximations are made in the implementation of the midpoint-distance profile method to overcome noise in the profile. First, a polynomial approximation is used to identify the regions near minima. Second, a binomial approximation is used to estimate the parameters of a component. These approximations are discussed below.

5.4.2. Location of Regions Near Minima

To determine the regions near a local minima, $\mathfrak{R}_i$, from the profile, a polynomial approximation is used. This polynomial, $\hat{p}(mp)$, is obtained by least-squares regression on the profile data. Figure 5-11 shows a profile and the corresponding polynomial approximation.

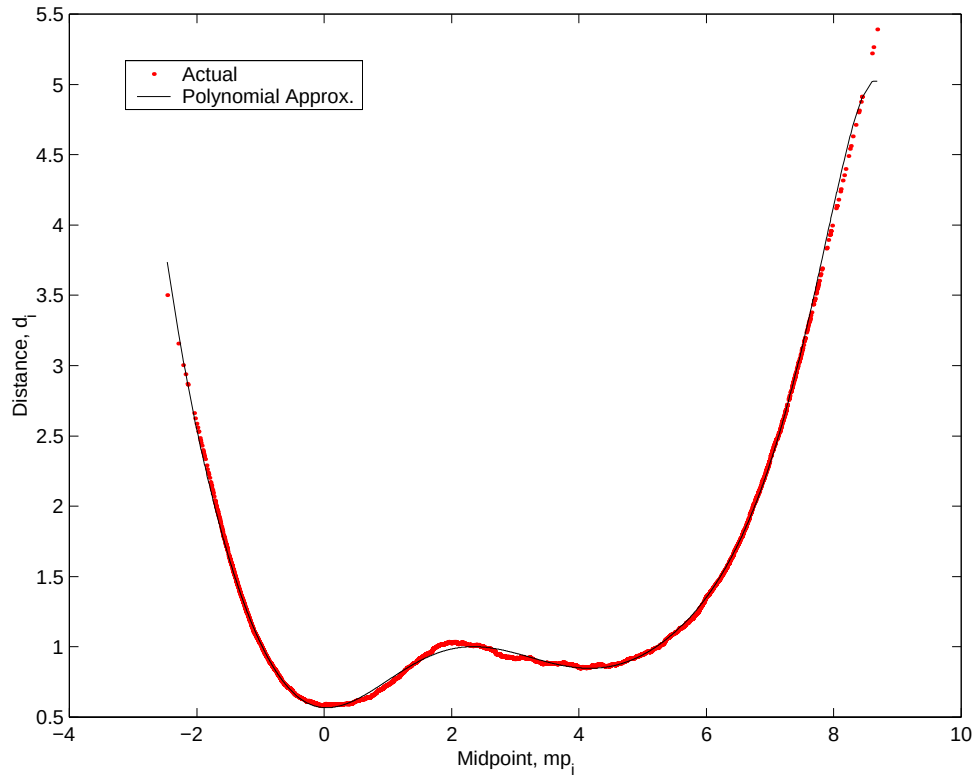


Figure 5-11 Polynomial approximation to the profile

Using the polynomial approximation, the stationary points can be found via Equation 5.28.

$$\frac{\partial}{\partial mp} \hat{p}(mp) = 0 \quad (5.28)$$

Using the second derivative, stationary points can be classified as minima, maximums, or inflection points. The region of influence of a minima, $\mathfrak{R}_i$, is bound by its closest inflection points. Figure 5-12 shows the regions of influence for the profile of Figure 5-11.

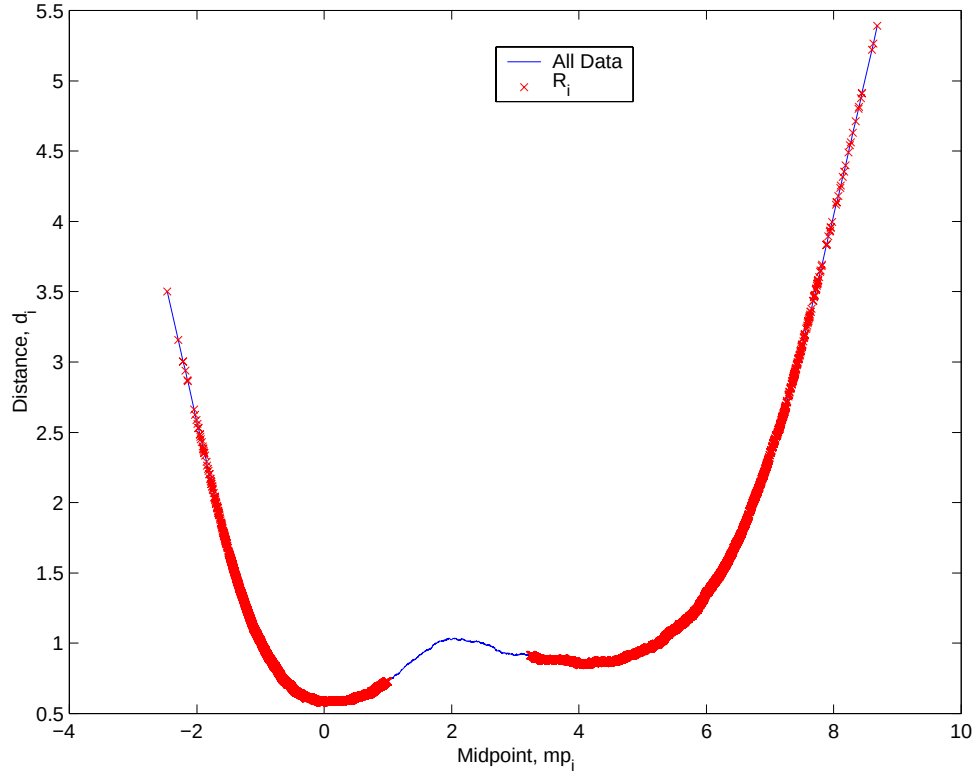


Figure 5-12 Regions of influence for a profile

Using the regions of influence, a binomial approximation is used to obtain the estimates of the parameters of a Gaussian component. This is discussed in the next section.

5.4.3. Estimation of Parameters

Using the region of influence of a minima, $\mathfrak{R}_i$, a binomial approximate is obtained using least-squares regression on data in the region. Equation 5.29 describes this approximation.

$$\hat{p}_i(mp) = a_i(mp)^2 + b_i(mp) + c_i \quad (5.29)$$

Figure 5-13 shows the two binomial approximations for the profile of Figures 5-11 and 5-12.

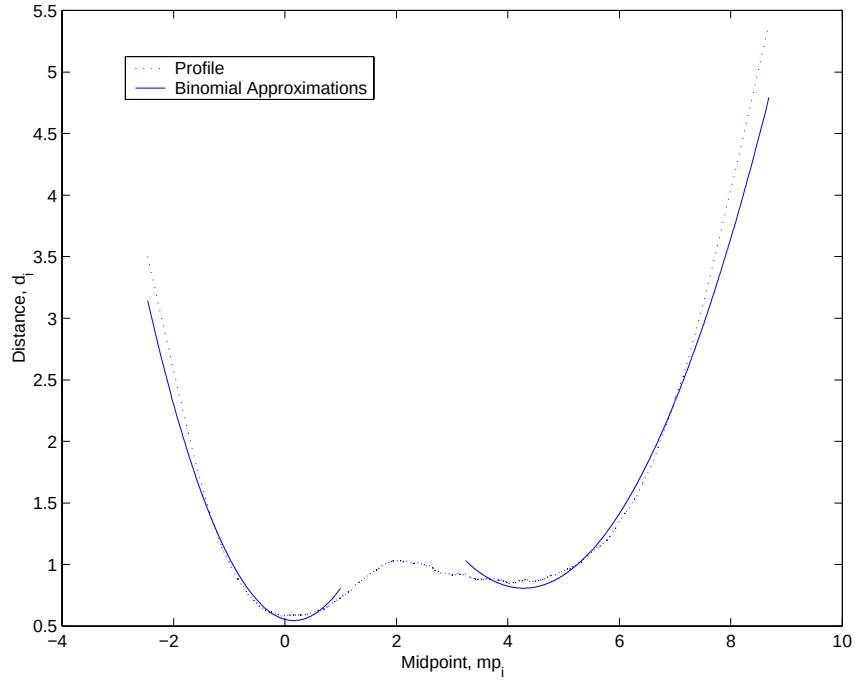


Figure 5-13 Binomial approximations in regions of influence

The estimate of location is obtained from the minima of the binomial approximation, as shown in Equation 5.30.

$$\hat{\mu}_i = \frac{-b_i}{2a_i} \quad (5.30)$$

Similarly, the estimate of scale can be obtained directly from the binomial coefficients and the correction factor discussed in Section 5.3.4.2. This estimate is shown in Equation 5.31.

$$\hat{\sigma}_i = \sqrt{\frac{d_{\min,i}(\delta(\alpha)-1)}{\alpha}} \quad (5.31)$$

$$d_{\min,i} = c_i - \frac{b_i^2}{4a_i}$$

Finally, estimates of weight are obtained using the location and scale estimates, as described in Equation 5.27.

5.4.4. Summary

This section described two approximations used to overcome the problem of noise that is present in the midpoint distance profile. These approximations allow determination of the estimates $\hat{\mu}_i$, $\hat{\sigma}_i$, and $\hat{w}_i$. The MATLAB source of this implementation is given in Appendix E.

5.5. Estimator Performance

Using the three estimation techniques of this chapter — method of moments (MOM) estimation, maximum likelihood estimation (MLE), and the midpoint-distance profile (MPDP) technique — a simulation experiment was conducted to assess the performance of the estimation methods. Appendix F discusses the simulation and provides a detailed analysis of the results. A summary of the results is presented below. The criteria discussed are estimator bias, rate of convergence, efficiency and numerical complexity. A discussion of the parametric sensitivity of each method is discussed. Finally, an analysis of the estimator extensibility is presented.

5.5.1. Bias

Equation 5.32 describes the bias of an estimator compared to the value of the parameter as observed in a finite data set. Using the observation parameter, rather than the “true” value, accounts for deviations due to sampling, e.g. the estimator’s bias is measured relative to the sample, rather than the population.

$$bias = \left| \hat{\theta} - \theta_{sample} \right| \quad (5.32)$$

Section F.2 provides detailed results from the simulation experiment. Table 5-2 highlights a key result, the average estimator bias compared to the sample parameter for the three estimation methods.

Table 5-2 Average Estimator Bias

	w_1, w_2	μ_1	μ_2	σ_1	σ_2
MLE	0.00417	0.02965	0.03682	0.01970	0.02671
MOM	0.01080	0.08108	0.09075	0.04619	0.05578
MPDP	0.03243	0.14692	0.06852	0.14401	0.54983

The data of Table 5-2 tends to indicate that both maximum likelihood estimation and the method of moments generate unbiased estimates. However, the results also indicate that the midpoint-distance profile does generate biased estimates. The average bias for MPDP estimation is larger than that of MOM and MLE, particularly for the scale estimates. Figures F-1 through F-6 indicate that the bias of MOM and MLE estimates tend toward zero as m increases; whereas the bias of MPDP estimates appear to approach a non-zero value. The most predominate bias is a tendency to underestimate the scale (see Appendix F). Figure 5-13 supports the conjecture that the bias is introduced from the polynomial approximations made in implementing the midpoint-distance profile method, as the profile is not truly a polynomial.

5.5.2. Rate of Convergence

As discussed in Section 2.3.1, estimators have a variance, as they are random variables. The variance of an estimator decreases exponentially as the sample size grows, e.g. $\sigma(\hat{\theta}) \propto m^{-b}$. The standard rate of convergence is $\frac{1}{2}$, whereas the midpoint of the shortest half converges more slowly, $b = \frac{1}{3}$.

The rates of convergence were estimated from the simulation experiment. Section F.3 discusses the estimation procedure. It should be noted that the estimates are not rigorous; the intent is to gain insight into the estimators' performance, not offer proof of their performance. Table 5-3 summarizes the results.

Table 5-3 Estimator Rates of Convergence

	w_1, w_2	μ_1	μ_2	σ_1	σ_2
MLE	0.4706	0.4884	0.4879	0.5045	0.4977
MOM	0.5618	0.5961	0.5115	0.5162	0.5366
MPDP	0.4484	0.4890	0.4695	0.4942	0.4926

The data of Table 5-3 tends to support the conclusion that all three estimation techniques converge at the standard rate, $\frac{1}{2}$. This result is expected, except in the case of the location estimation of the midpoint-distance profile, which is a generalization of the midpoint of the shortest half, which converges at a rate of $\frac{1}{3}$. A possible explanation of the faster than expected convergence is the polynomial approximation used to find the

locations, rather than the actual minimum value of the region. The conjecture is that the polynomial converges faster than the true minimum. This is plausible, since the polynomial is estimated using least-squares regression.

5.5.3. Efficiency

Estimator efficiency is a measure of the variance of an estimator. Relative efficiency ranks one estimator against another. As discussed in Section 2.3.1, consistent maximum likelihood estimators are asymptotically efficient; therefore, the other two estimation methods were compared to it to obtain relative efficiencies. As with the rates of convergence, these figures are weak estimates, intended to provide insight to the performance of the estimators, not rigorous ascertains about their performance. Table 5-4 summarizes the observed relative efficiencies.

Table 5-4 Estimator Efficiency Relative to Maximum Likelihood

	w_1, w_2	μ_1	μ_2	σ_1	σ_2
MOM	0.5910	0.5582	0.5658	0.6345	0.6493
MPDP	0.8028	0.7235	0.7850	0.5256	0.6073

Table 5-4 shows that both the method of moments and the midpoint-distance profile are not as efficient as maximum likelihood estimation. For weights and locations, the midpoint-distance profile is more efficient than the method of moments; however, the method of moments outperforms the midpoint-distance profile in terms of scale estimation.

5.5.4. Numerical Complexity

The complexity of each technique was assessed as a measure of algorithm scalability. The average number of floating point operations (flops) required for each method to obtain a set of estimates was observed as a function of m . Figure 5-14 shows the results. The method of moments and maximum likelihood scale at the same rate; however, the method of moments requires fewer numerical operations. The midpoint-distance profile is less scalable.

Since a numerical minimization technique was used for maximum likelihood, an initial guess is required. The method of moments estimate was used. However, the flops

results for maximum likelihood estimation represent only the numerical search. A truer estimate would include the overhead of calculating the initial guess; however, since the method of moments is about two orders of magnitude more efficient than maximum likelihood estimation, the impact of that overhead is minimal.

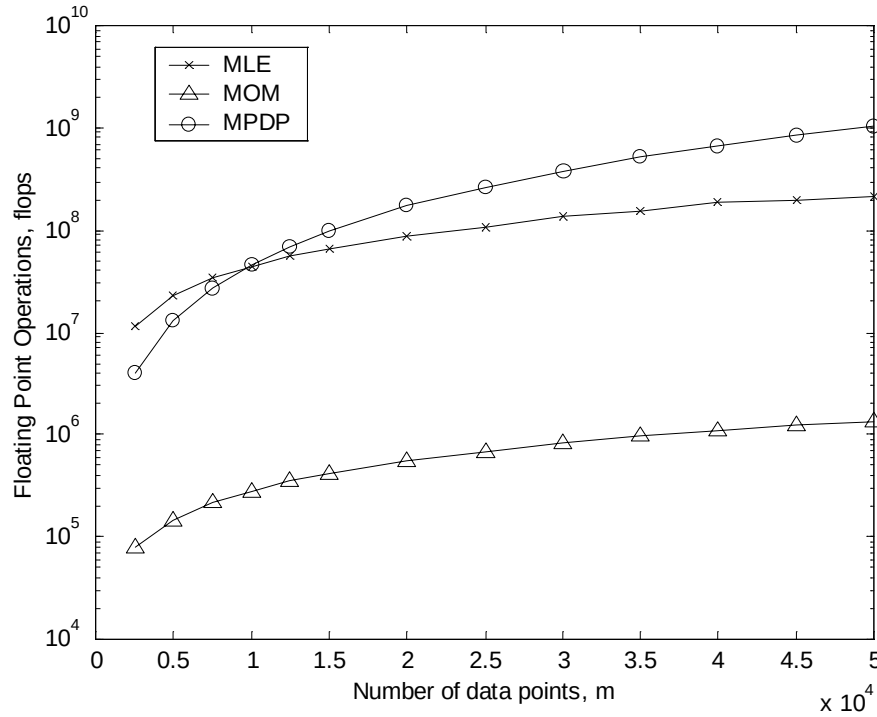


Figure 5-14 Numerical Complexity

5.5.5. Parametric Sensitivity and Effects

The method moments requires parameters (other than the data set) for estimation. However, maximum likelihood estimation requires an initial guess and the midpoint-distance profile requires the selection of the parameter α . The effect of the selection of these parameters is discussed below.

As mentioned in Section 5.5.4, the numerical search implementation of maximum likelihood estimation requires an initial guess. If the objective function of the numerical search is not convex, the search may converge to a local, rather than a global, minimum. In general, objective functions of maximum likelihood estimation are not convex. This underscores the importance of the selection of a “good” initial guess.

The midpoint-distance profile requires the parameter α to be defined. This parameter represents the window of data used in developing the midpoint-distance profile. Section 5.3.4.2 mentions that a detectable component's weight lies in the interval $\{\alpha, 1\}$. Therefore, α represents the minimum detectable component weight of the technique.

5.5.6. Extensibility

The previous section relied on simulation results to assess the bias, rate of convergence, efficiency, and complexity of the three estimation techniques. This section addresses the extensibility, or the ability to estimate the parameters of n -component Gaussian mixtures, rather than 2-component mixtures, of the three methods. The section analyzes the nature of the techniques, rather than utilizing simulation results.

While extensible in theory, the method of moments is not extensible in practice. An n -component mixture has $3n - 1$ parameters and the first $3n - 1$ analytic moments could be equated to the empirical moments to obtain a system of equations to solve. Since the system of equations is non-linear, a technique for solving the system, analogous to that of Section 5.2.2, would be required.

To extend maximum likelihood estimation to n components, the log-likelihood function of Equation 5.10 would need to be extended to accommodate the n components. This increases the dimensionality of the search space of the numerical minimization routine to $3n - 1$, implying a slower convergence. Moreover, since the method of moments does not extend, obtaining initial guesses for the routine is problematic.

The midpoint-distance profile is inherently extensible. Additional components appear as additional minima on the profile. Estimates of additional components are obtained in the same manner as the first two components. Figure 5-15 demonstrates this extensibility, showing the profile ($\alpha = 0.05$) of 15,000 observations from a three-component Gaussian mixture with the following parameters.

$$\begin{Bmatrix} w_1 & \mu_1 & \sigma_1 \\ w_2 & \mu_2 & \sigma_2 \\ w_3 & \mu_3 & \sigma_3 \end{Bmatrix} = \begin{Bmatrix} 0.2 & -4 & 2 \\ 0.4 & 0 & 1 \\ 0.4 & 2 & 0.5 \end{Bmatrix}$$

The estimates obtained from the midpoint distance profile technique are as follows.

$$\begin{Bmatrix} \hat{w}_1 & \hat{\mu}_1 & \hat{\sigma}_1 \\ \hat{w}_2 & \hat{\mu}_2 & \hat{\sigma}_2 \\ \hat{w}_3 & \hat{\mu}_3 & \hat{\sigma}_3 \end{Bmatrix} = \begin{Bmatrix} 0.182 & -4.12 & 1.61 \\ 0.432 & -0.06 & 0.984 \\ 0.386 & 2.00 & 0.362 \end{Bmatrix}$$

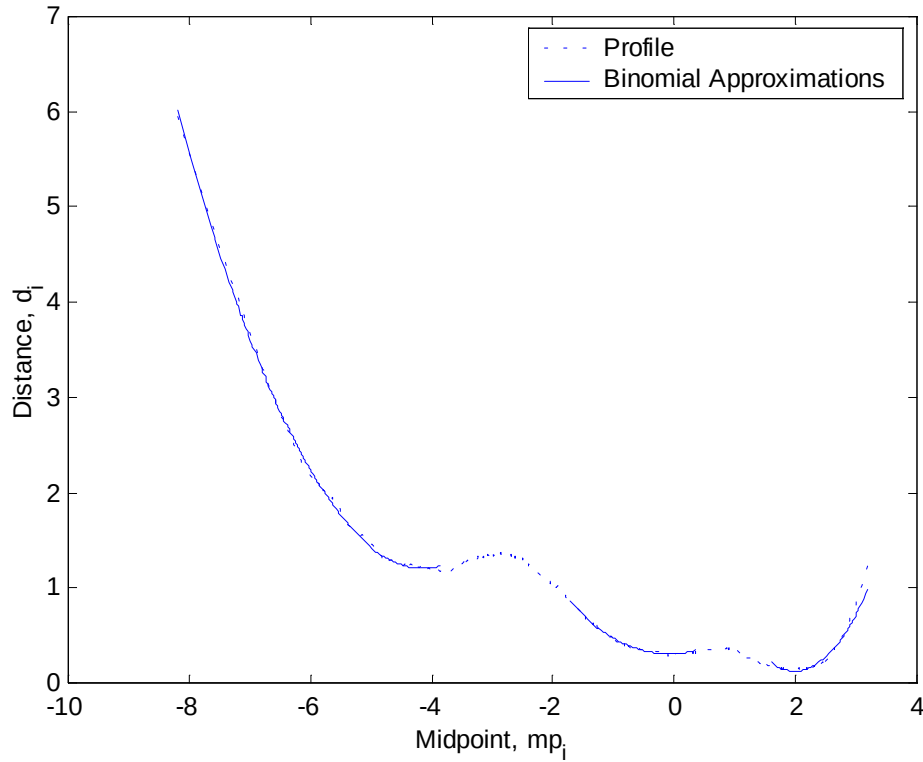


Figure 5-15 Profile of a three component mixture

5.6. Application to WAN Modeling

The previous sections have analyzed three multimodal estimation techniques in a generic fashion. As stated in Section 5.1, the motivation for this study is its application to WAN modeling. This section returns to this application realm by revisiting the NNTP duration observations previously discussed.

The observations of NNTP connection duration were multimodal, with remote (typical) connections favoring the larger mode. Figure 5-16 shows an estimate of the probability density obtained from the data, and the probability densities of three Gaussian mixtures fit using the three techniques of this chapter.

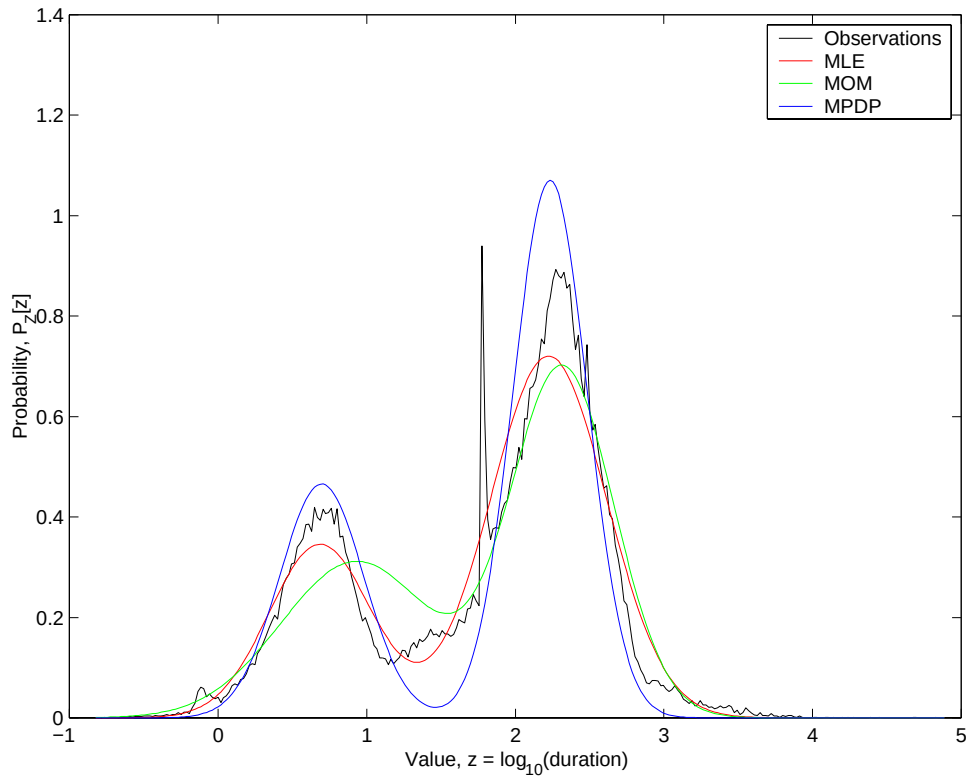


Figure 5-16 Gaussian mixture fits to NNTP data

Figure 5-16 shows several interesting trends. The estimation of the smaller mode is highly influenced by the cluster of data located near $z = 1.8$; this contamination attracts the smaller mode. Both maximum likelihood and the method of moments appear to have under-weighted the larger mode, possibly due to the effect of this cluster of data. The midpoint-distance profile estimation tends to reject this cluster and isolate the probability mass near the modes. These trends tend to indicate a higher level of robustness to inliers of the midpoint-distance profile estimation method.

The bias of the midpoint-distance profile as discussed in Section 5.5.1 gives rise to a drawback when the larger mode is isolated as a model for the remote connections. The component is too narrow when isolated because the midpoint-distance profile tends to underestimate the scale. Figure 5-17 illustrates this effect.

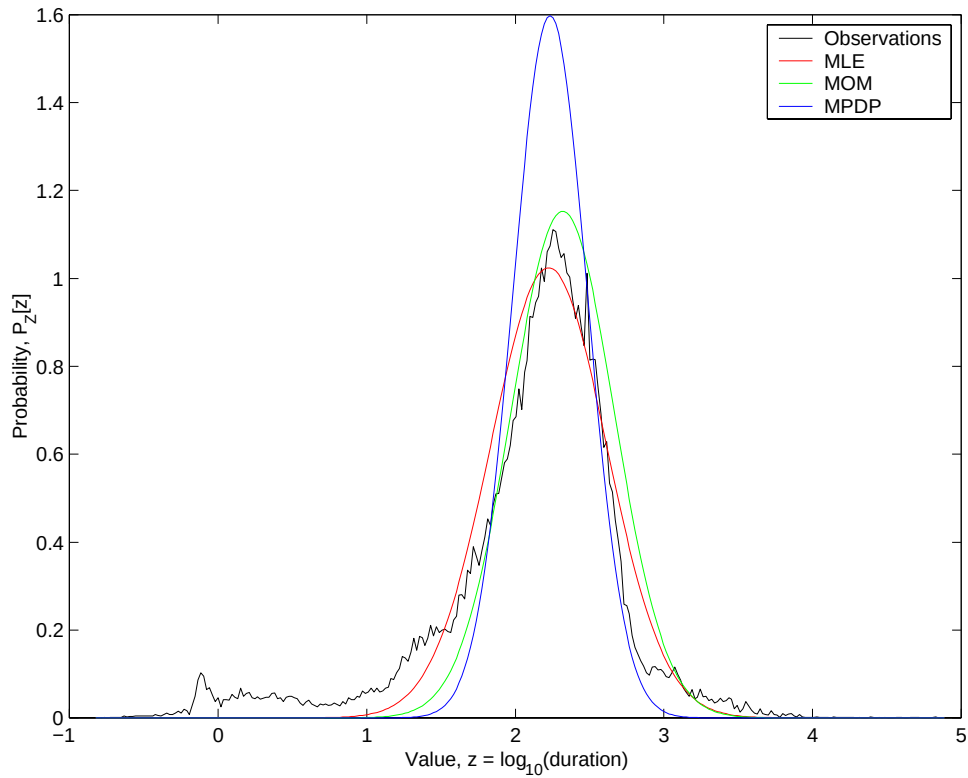


Figure 5-17 Gaussian component as a model for remote NNTP connections

Using the NNTP duration data, the three estimation techniques were evaluated in the context of WAN modeling. The inferences of this analysis are twofold. Compared to maximum likelihood estimation and the method of moments, the midpoint-distance profile tends to isolate the modes. This behavior can be beneficial if the data is subject to contamination from inliers; however, when components are isolated the midpoint-distance profile fit suffers from a bias toward underestimation of the scale.

5.7. Summary

This chapter began with a review of the problem of multimodal density estimation in the context of WAN traffic modeling. The two-component Gaussian mixture was used as a model for a bimodal distribution. Two conventional estimation techniques — the method of moments and maximum likelihood estimation — were reviewed. The theory and implementation of a new estimation technique, the midpoint-distance profile, were presented. The three estimation techniques were evaluated via a simulation study and

using NNTP traffic observed in previous network traffic studies. The key results are summarized below:

- Midpoint-distance profile estimation appears to generate biased estimates. In particular, scale estimates appear to be underestimated.
- Midpoint-distance profile location estimates converge faster than expected. This is likely due to approximations made in the implementation.
- Maximum likelihood estimation is more efficient than the other two techniques.
- Midpoint-distance profile estimation is less scalable than the other two methods. However, for smaller data sets, $m \lesssim 10,000$, midpoint-distance estimation is more computationally feasible than maximum likelihood estimation.
- Midpoint-distance profile estimation is inherently extensible to distributions with more than two modes. Extending the other techniques is problematic.
- Probability densities fit using the midpoint-distance profile technique tend to isolate the modes and reject contaminating inliers.
- Modeling a single component using the midpoint-distance estimates suffers from the technique's tendency to underestimate the scale.

Chapter 6: Conclusion

This thesis has developed and presented solutions to the problem of modeling wide-area network traffic from observations contaminated with atypical data. Prior network traffic studies have dealt with atypical traffic — traffic that is particular to the observation site and not generally anticipated — in an ad-hoc manner. A technique for detecting atypical traffic is developed and studied. This technique was validated against data from the literature and random simulation data. The problem of bimodal probability distribution estimation is examined in general, and then within the context of atypical traffic detection. Two putative techniques, the method of moments and maximum likelihood estimation, are reviewed. A novel method, the midpoint-distance profile, based on estimators from robust statistics is developed. The advantages and disadvantages of these techniques are examined using simulation experiments and traffic observations.

This concluding chapter highlights each of the previous chapters, and then postulates some avenues for future research.

6.1. *Summary*

Chapter 2 reviewed the various approaches to network traffic modeling in the literature and the bimodal nature of WAN traffic. Several statistical concepts were reviewed, including the Gaussian mixture distribution, properties and techniques of statistical estimation, measuring quality of fit, and robust estimation. The concepts reviewed in Chapter 2 were exploited in various later chapters.

Chapter 3 explored the nature of atypical traffic. Using previously studied WAN observations, three major sources of atypical traffic were identified, as follows.

- Ill-behaved user-level applications that create atypical usage patterns in the TCP/IP protocol they use.
- Abnormal circumstances that create errors within a TCP/IP application-layer protocol.
- Location specific topologies or configurations that create atypical protocol behavior.

Using the concept of quality of fit, parsimonious analytical modeling of WAN traffic was found to benefit from the removal of atypical traffic.

A technique for automated detection and modeling of atypical traffic was presented in Chapter 4. This technique relies on the empirical distribution function of a data set. The technique was found to be effective in detecting the atypical traffic of previously studied observations and randomly generated data sets. The technique was further found to be highly scalable.

Chapter 5 conceptualized the problem of multimodal probability distribution modeling in the field of atypical WAN traffic. Two conventional estimation techniques, the method of moments and maximum likelihood estimation, were reviewed for the two-component (bimodal) Gaussian mixture distribution. A new technique, the midpoint-distance profile, was developed. The properties of these three techniques were evaluated via simulation and using WAN observations. Each of the estimation techniques had advantages and disadvantages.

6.2. *Future Work*

Chapters 4 and 5 both present new techniques for a given problem, as such, they are full of opportunities for further research. A few ideas are presented below.

The contamination detection technique of Chapter 4 represents a specialized, context-specific technique for inlier detection. Although the method of Chapter 4 was shown to be effective, applying techniques from unsupervised learning, such as clustering, to the atypical traffic detection problem could prove interesting.

On a more detailed level, the algorithm of Chapter 4 most likely would benefit from a thorough optimization study to improve the performance and scalability of the algorithm. The technique could be studied in the context of other network protocols, such as HTTP, or in other application areas subject to strong inlier contamination.

The midpoint-distance profile estimation method presented in Chapter 5 exploits concepts from robust statistics. However, a robustness study of this technique to the other estimation methods was not performed. Exploring the effect of outliers, atypically large valued observations, on the estimation techniques is an obvious avenue of research. Network modeling does not lend itself to this area of research. Network traffic is often

heavy-tailed; therefore, conventional “outliers” often represent interesting and important data in network traffic modeling. This, by no means, trivializes the importance of outlier robustness in other fields.

The treatment of the midpoint-distance profile method in Chapter 5 was limited to that which was feasible in a thesis. Certainly more insight into the technique and its applications could be gained by a more thorough study.

References

- [1] V. Frost and B. Melamed, "Traffic Modeling for Telecommunications Networks," *IEEE Communications Magazine*, vol. 32, no. 3, pp. 70-81, March 1994.
- [2] H. Michiel and K. Laevens, "Teletraffic Engineering in a Broad-Ban Era," *Proceedings of the IEEE*, vol. 85, no. 12, Dec., pp. 2007-2033, 1997.
- [3] P. Danzig, S. Jamin, R. Cáceres, D. Mitzel, and D. Estrin, "An Empirical Workload Model for Driving Wide-area TCP/IP Network Simulations," *Journal of Internetworking*, vol. 3, no. 1, pp. 1-26, Mar. 1992.
- [4] P. Danzig and S. Jamin, "tcplib: A Library of TCP Internetwork Traffic Characteristics", University of Southern California, Technical Report, USC-CS-91-495, 1991.
- [5] V. Paxson, "Emperically Derived Models of Wide-Area TCP Connections," *IEEE/ACM Transactions on Networking*, vol. 2, no. 4, pp. 316-336, August 1994.
- [6] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1-15, Feb 1994.
- [7] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226-244, June 1995.
- [8] J. Beran, R. Sherman, M. Taqqu, and W. Willinger, "Long-Range Dependence in Variable Bit-Rate Traffic," *IEEE Transactions on Communications*, vol. 43, no. 2-4, pp. 1566-1579, Feb.-Apr. 1994.
- [9] V. Paxson, "Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic," *Computer Communication Review*, vol. 27, no. 5, pp. 5-18, Oct. 1997.
- [10] A. C. Klaiber, "Architectural support for compiler-generated data-parallel programs," dissertation, University of Washington, 1994.
- [11] N. R. McKenzie, "The Cranium Network Interface Architecture: Support for Message Passing on Adaptive Packet Routing Networks," dissertation, University of Washington, 1997.

- [12] R. F. Smith II, "Data Traffic Modeling for Asynchronous Transfer Mode Networks", dissertation, Tulane University, 1998.
- [13] P.C. Wong and T.S. Yum, "An Integrated Services Token-Controlled Ring Network," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 5, pp. 670-679, June 1989.
- [14] P.C. Wong and T. S. Yum, "Design and Analysis of a Pipeline Ring Protocol," *IEEE Transactions on Communications*, vol. 42, no. 2-4, pp. 1153-1161., Feb.-Apr. 1994.
- [15] N. L. Johnson and S. Kotz, *Continuous Univariate Distributions*. New York: John Wiley and Sons, 1970.
- [16] K. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications (Second Edition)*. New York: John Wiley and Sons, 2002.
- [17] T. Cover and J. Thomas, *Elements of Information Theory*, New York: John Wiley and Sons, 1991.
- [18] R. Grübel, "The Length of the Shorth," *Annals of Statistics*, vol. 16, no. 2, pp. 619-628, Jun. 1988.
- [19] J. Milton and J. Arnold, *Introduction to Probability and Statistics*. New York: McGraw-Hill, 1995.
- [20] J. W. Pratt, "F.Y. Edgeworth and R.A. Fisher on the Efficiency of Maximum Likelihood Estimation," *Annals of Statistics*, vol. 4, no. 3, pp. 501-514, May 1976.
- [21] R. Jain, *The Art of Computer Systems Performance Analysis*. New York: John Wiley and Sons, 1991.
- [22] A. Law and W. Kelton, *Simulation Modeling and Analysis*. New York: McGraw-Hill, 1991.
- [23] R. D'Agostino and M. Stephens, Eds., *Goodness-of-Fit Techniques*. New York: Marcel Dekker, Inc., 1986.
- [24] S. Pederson and M. Johnson, "Estimating Model Discrepancy," *Technometrics*, vol. 32, no. 3, pp. 305-314, Aug. 1990.
- [25] D. Scott, "On Optimal and Data-Based Histograms," *Biometrika*, vol. 66, no. 3, pp. 605-610, Dec. 1979.

- [26] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, New York: John Wiley and Sons, 1986.
- [27] O. Nasraoui, "Object and Relational Clustering Based on New Robust Estimators and Genetic Niching with Applications to Web Mining," dissertation, University of Missouri-Columbia, 1999.
- [28] D.F. Andrews, P.J. Bickel, F.R. Hampel, P.J. Huber, W.H. Rogers, and J.J. Tukey, *Robust Estimation of Location: Survey and Advances*. Princeton University Press, Princeton, N.J., 1972.
- [29] P. J. Rousseeuw, "Least median of squares regression", *Journal of the American Statistical Association*, vol. 79, no. 333, pp. 871-880, Dec. 1984.
- [30] J. D. Vest, "Robust, Location-Free Scale Estimators for the Linear Regression and k-Sample Models," dissertation, Virginia Polytechnic Institute and State University, 1996.
- [31] V. Paxson, "Growth Trends in Wide-Area TCP Connections," *IEEE Network*, vol. 8, no. 4, pp. 8-17, Jul. - Aug. 1994.
- [32] B. Kantor and P. Lapsley, "Network News Transfer Protocol: A Proposed Standard for Stream-Based Transmission of News," RFC 977, Feb. 1986, Available HTTP: <http://www.ietf.org/rfc/rfc0997.txt>
- [33] J. Postel, "Simple Mail Transfer Protocol", RFC 821, Aug. 1982, Available HTTP: <http://www.ietf.org/rfc/rfc0821.txt>
- [34] T. Berners-Lee, R. Feilding, and H. Frystyk, "Hypertext Transfer Protocol – HTTP/1.0", RFC 1945, May 1996, Available HTTP: <http://www.ietf.org/rfc/rfc1945.txt>
- [35] "The Apache Software Foundation," [Online document], Available HTTP: <http://www.apache.org/>
- [36] "CERT® Incident Note IN-2001-10," [Online document], Aug. 2001, Available HTTP: http://www.cert.org/incident_notes/IN-2001-10.html
- [37] "CERT® Advisory CA-2001-26 Nimda Worm," [Online document], Sept. 2001, Available HTTP: <http://www.cert.org/advisories/CA-2001-26.html>
- [38] P. Danzig, J. Mogul, V. Paxson, and M. Schwartz, "The Internet Traffic Archive," [Online document], Apr. 2000, Available HTTP: <http://ita.ee.lbl.gov/>

- [39] J.C. Lagarias, J.A. Reeds, M.H. Wright, and P.E. Wright, "Convergence Properties of the Nelder-Mead Simplex Algorithm in Low Dimensions", *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 1122-1147, 1998.

Appendix A: WAN Connection Measurement EDFs

This appendix shows the empirical distribution functions (EDFs) for the three traffic protocols discussed in Section 3.3. The EDFs for the three variables: duration, bytes transmitted, and bytes received.

A.1. Telnet Traffic

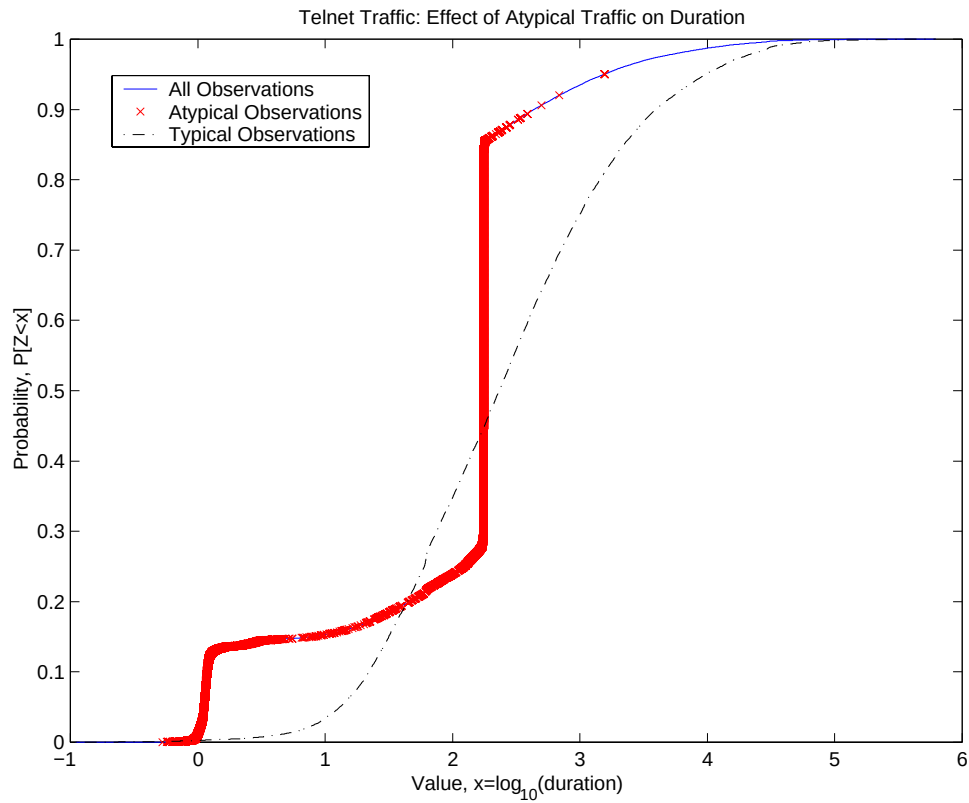


Figure A-1 Effect of atypical traffic on telnet duration

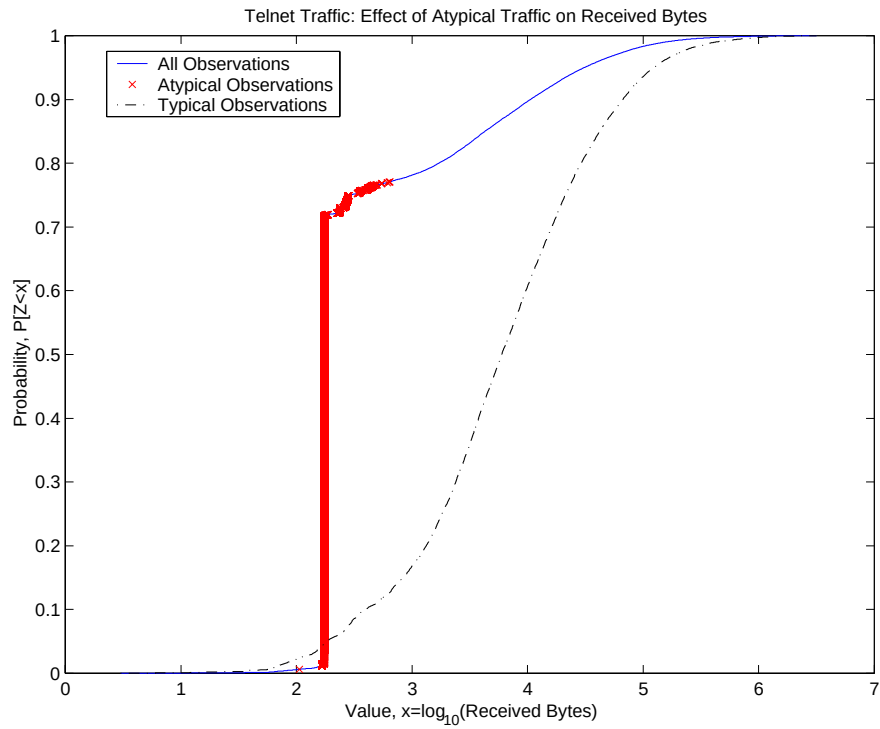


Figure A-2 Effect of atypical traffic on telnet received bytes

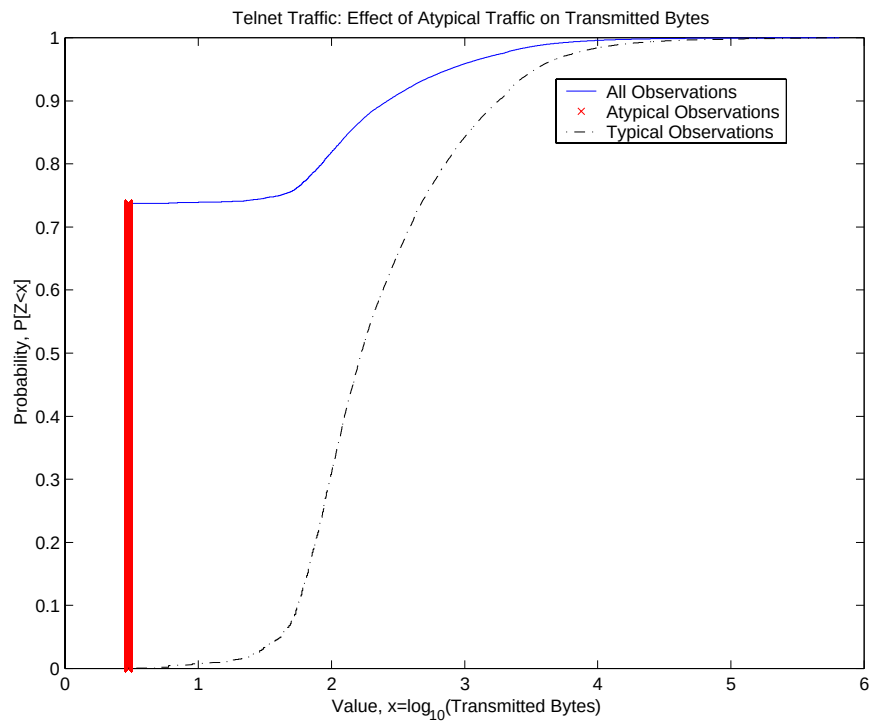


Figure A-3 Effect of atypical traffic on telnet transmitted bytes

A.2. NNTP Traffic — Effect of Atypical Traffic

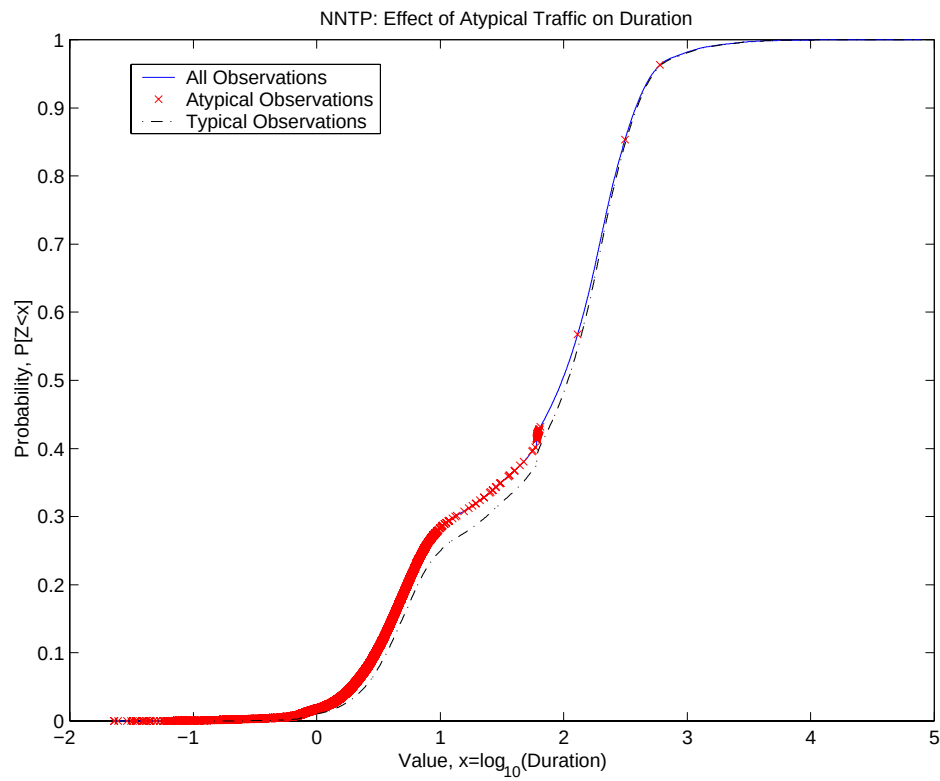


Figure A-4 Effect of atypical traffic on NNTP duration

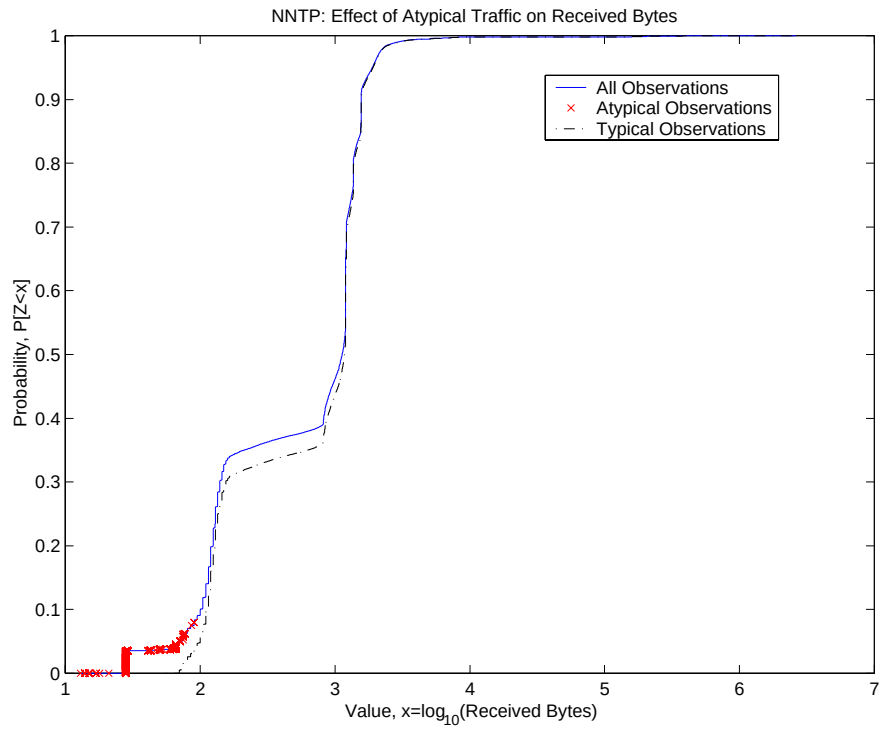


Figure A-5 Effect of atypical traffic on NNTP received bytes

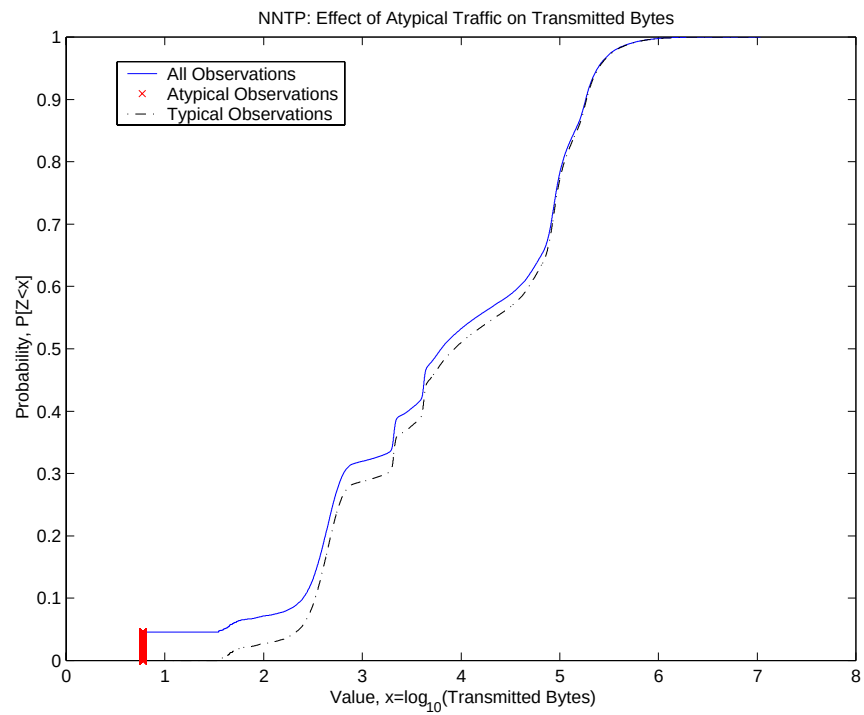


Figure A-6 Effect of atypical traffic on NNTP transmitted bytes

A.3. NNTP Traffic — Effect of Co-location

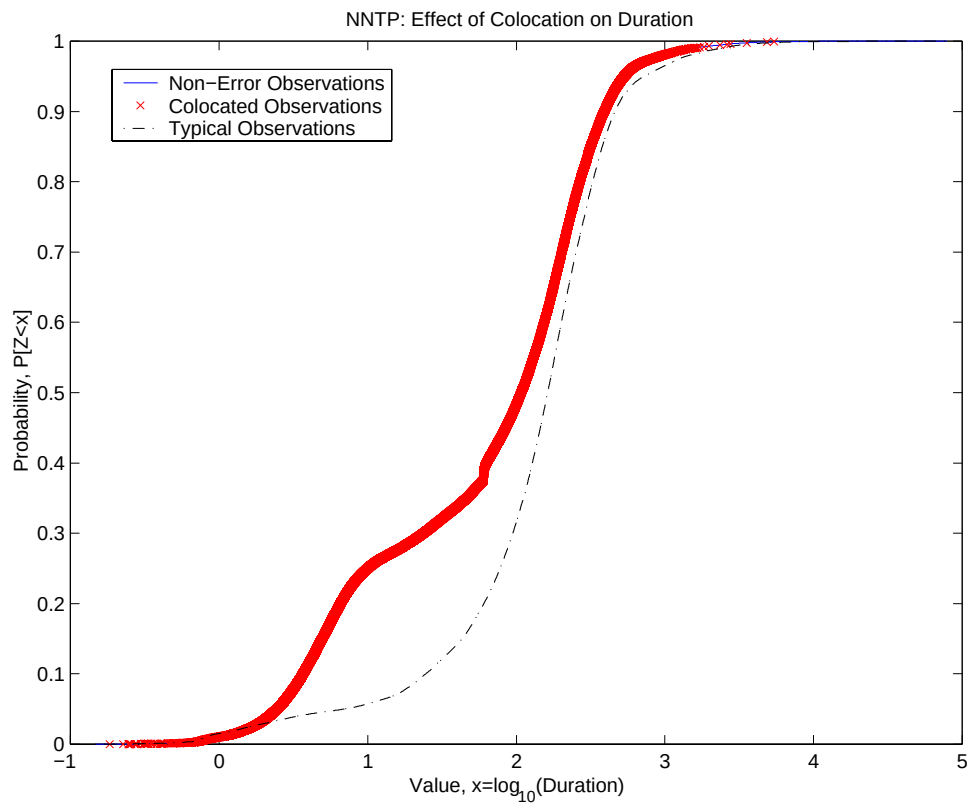


Figure A-7 Effect of co-location on NNTP duration

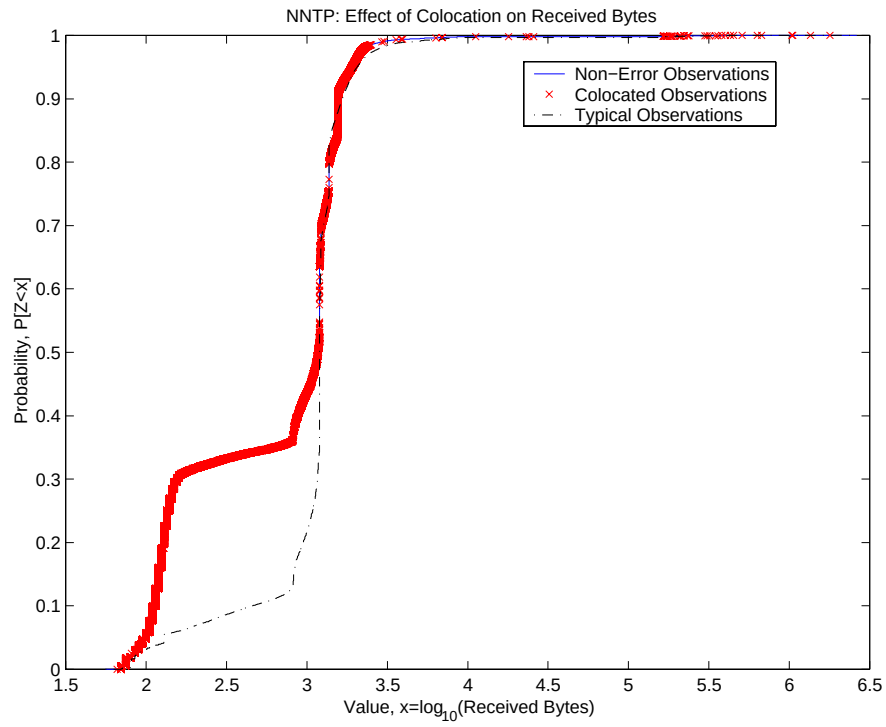


Figure A-8 Effect of co-location on NNTP received bytes

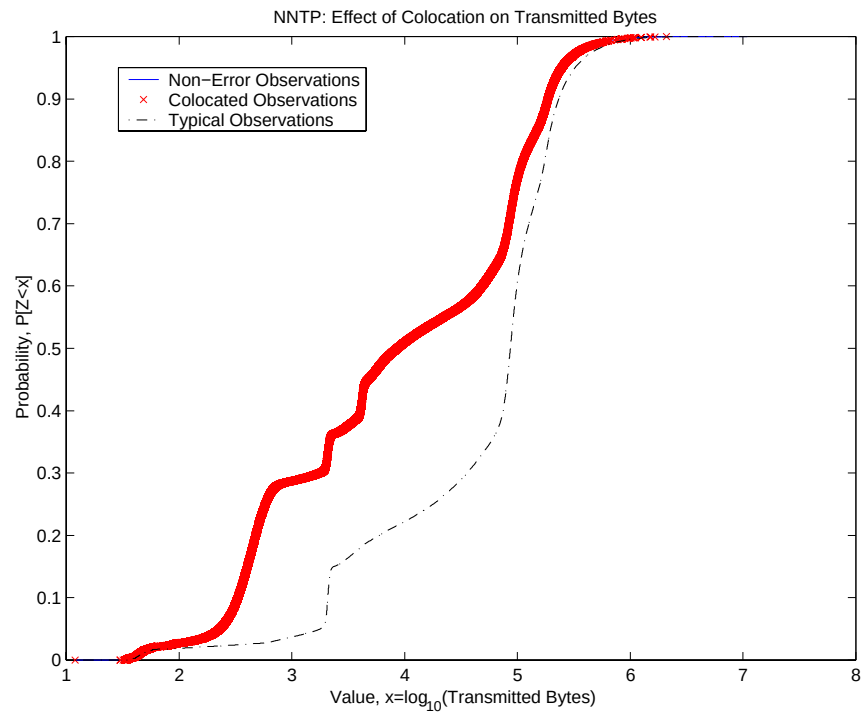


Figure A-9 Effect of co-location on NNTP transmitted bytes

A.4. SMTP Traffic

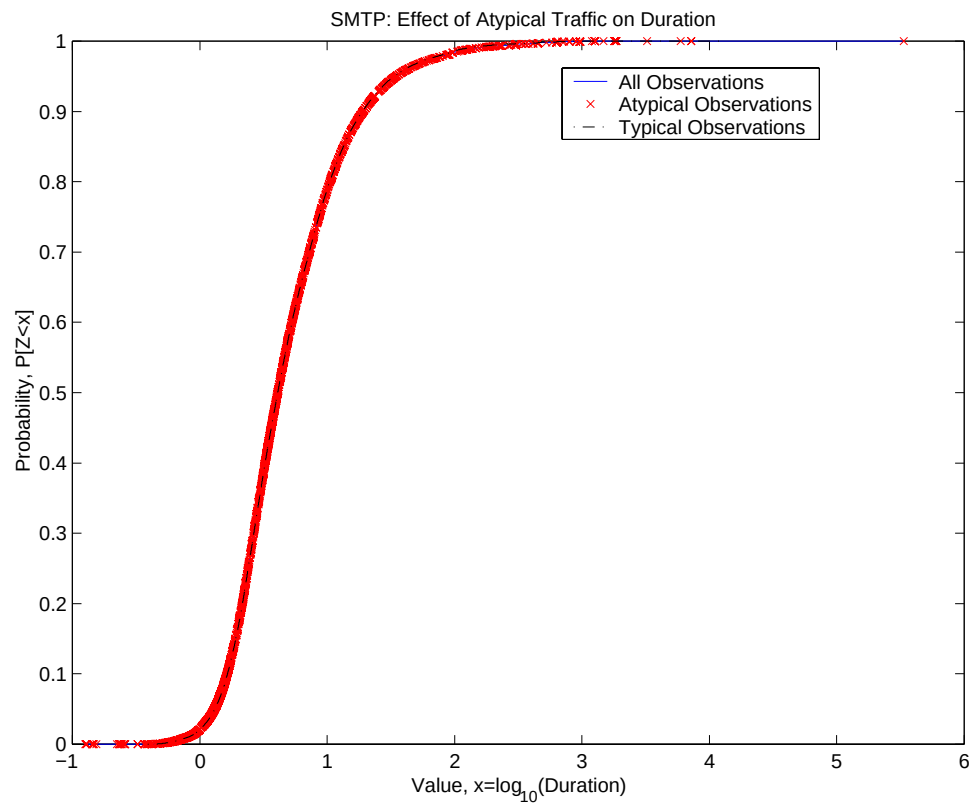


Figure A-10 Effect of atypical traffic on SMTP duration

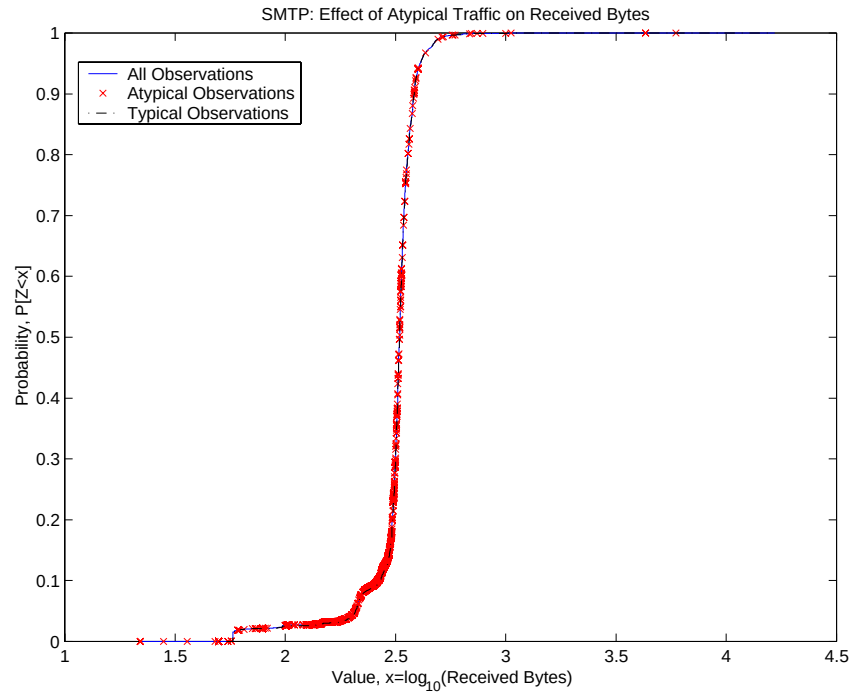


Figure A-11 Effect of atypical traffic on SMTP received bytes

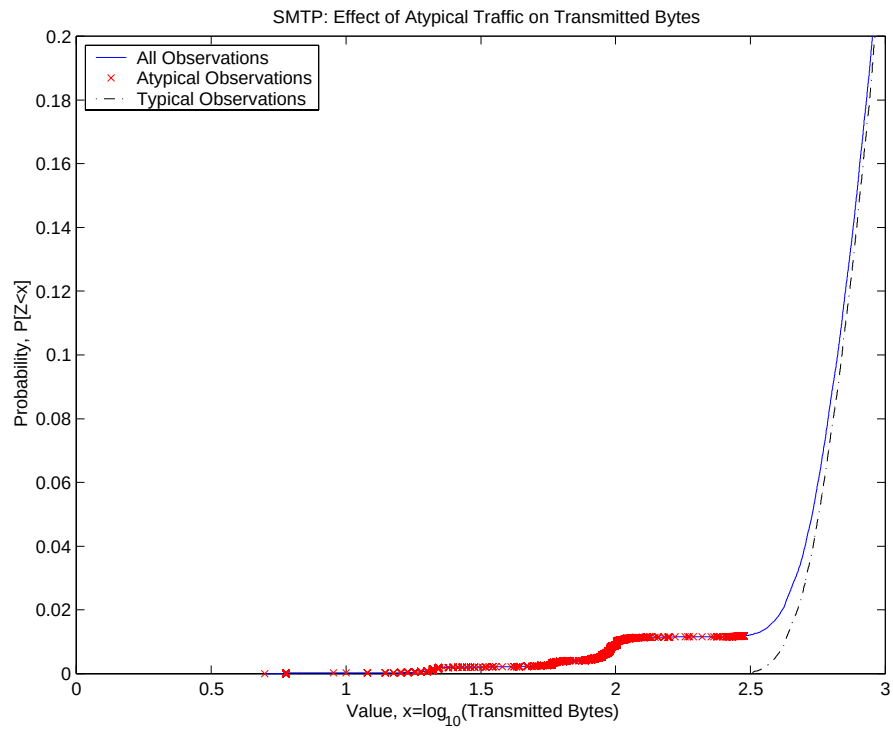


Figure A-12 Effect of atypical traffic on SMTP transmitted bytes

Appendix B: Model Merging Technique

Section 4.3.3 presented the augmented coverage matrix reduction approach to model merging. This appendix presents an example of the method and two examples that demonstrate the impact of the choice of the method's arguments to the overall stability of the contamination algorithm presented in Chapter 4.

B.1. Example of Model Merging Technique

Consider a single-variate dataset and the following hypothesized contamination models:

$$\begin{aligned} C_1 &= \{0,1,2,3,4\} \\ C_2 &= \{0,1\} \\ C_3 &= \{1,2\} \\ C_4 &= \{4,5\} \end{aligned} \tag{B.1}$$

Using Equations 4.11 and 4.13, the augmented coverage matrix can be expressed as:

$$ACM = \left[\begin{array}{cccc|c} 1 & 0.4 & 0.4 & 0.2 & 1 \\ 1 & 1 & 0.5 & 0 & 2 \\ 1 & 0.5 & 1 & 0 & 3 \\ 0.5 & 0 & 0 & 1 & 4 \end{array} \right] \tag{B.2}$$

Using the method described in Section 4.3.3, the largest off-diagonal value of the coverage matrix is found. In this case, that corresponds to the second row and first column, $\psi_{i,j}^{\max} = \psi_{2,1} = 1$. Assuming $\gamma = 0.9$ then $\psi_{i,j}^{\max} > \gamma$. Therefore, the second row and column can be removed from the augmented coverage matrix, as C_1 subsumes C_2 . This gives the augmented coverage matrix below:

$$\begin{aligned} ACM &= \left[\begin{array}{cccc|c} 1 & 0.4 & 0.4 & 0.2 & 1 \\ 1 & 1 & 0.5 & 0 & 2 \\ 1 & 0.5 & 1 & 0 & 3 \\ 0.5 & 0 & 0 & 1 & 4 \end{array} \right] \\ &= \left[\begin{array}{ccc|c} 1 & 0.4 & 0.2 & 1 \\ 1 & 1 & 0 & 3 \\ 0.5 & 0 & 1 & 4 \end{array} \right] \end{aligned} \tag{B.3}$$

Continuing the process yields $\psi_{i,j}^{\max} = \psi_{2,1} = 1$. Again, $\psi_{i,j}^{\max} > \gamma$; correspondingly C_1 subsumes C_3 . This is represented below:

$$\begin{aligned} ACM &= \left[\begin{array}{ccc|c} 1 & 0.4 & 0.2 & 1 \\ 1 & 1 & 0 & 3 \\ 0.5 & 0 & 1 & 4 \end{array} \right] \\ &= \left[\begin{array}{cc|c} 1 & 0.2 & 1 \\ 0.5 & 1 & 4 \end{array} \right] \end{aligned} \quad (B.4)$$

Using this augmented coverage matrix and repeating the process yields $\psi_{i,j}^{\max} = \psi_{2,1} = 0.5$. However, $\psi_{i,j}^{\max} < \gamma$; therefore, the model merging process ends. The remaining (non-subsumed) models can be determined from the final column of the augmented matrix. In this case those models are $C = \{C_1, C_4\}$.

B.2. Instability of Merging on All Models

Section 4.3.3 alludes to the instability of the contamination model detection algorithm if the merging is based on all models, rather than just the hypothesized models for a given iteration, e.g., $f(C, D, \gamma)$ instead of $f(C^H, D, \gamma)$. This section will outline a set of conditions that can lead to this instability and verify these conditions with an example.

Consider a dual-variate dataset, D , with the contamination models $C_1 \ni \{x_1 = A\}$ and $C_2 \ni \{x_2 = B\}$. Let x_1 and x_2 be independent unit normal variates, but for the contamination. The conditions for instability of the contamination detection process are stated below.

$$\begin{aligned} I. \quad & P[x_1 = A] > \alpha \\ II. \quad & P[x_2 = B] > \alpha \\ III. \quad & P[x_1 = A | x_2 = B] > \gamma \\ IV. \quad & P[x_2 = B | x_1 \neq A] > \alpha \end{aligned} \quad (B.5)$$

The first condition causes C_1 to be hypothesized at the first iteration; similarly, the second condition causes C_2 to be hypothesized at the same iteration. The third condition causes C_1 to subsume C_2 during the first iteration's model merging stage. The final condition ensures that C_2 will reappear during the successive iteration.

These conditions can be verified via an example. Consider the following distribution of $\{x_1, x_2\}$:

$$\begin{aligned}
 P[x_1 = A, x_2 = B] &= 0.5 \\
 P[x_1 = A, x_2 \neq B] &= 0.4 \\
 P[x_1 \neq A, x_2 = B] &= 0.08 \\
 P[x_1 \neq A, x_2 \neq B] &= 0.02
 \end{aligned} \tag{B.6}$$

Assume that $\gamma = 0.85$, $\alpha = 0.1$, and θ_c is sufficient to detect the contamination regions. The conditions of Equation B.6 are satisfied, as shown below.

$$\begin{aligned}
 I. \quad P[x_1 = A] &= P[x_1 = A, x_2 = B] + P[x_1 = A, x_2 \neq B] \\
 &= 0.5 + 0.4 = 0.9
 \end{aligned}$$

$$P[x_1 = A] > \alpha$$

$$\begin{aligned}
 II. \quad P[x_2 = B] &= P[x_1 = A, x_2 = B] + P[x_1 \neq A, x_2 = B] \\
 &= 0.5 + 0.08 = 0.58
 \end{aligned}$$

$$P[x_2 = B] > \alpha$$

$$\begin{aligned}
 III. \quad P[x_1 = A | x_2 = B] &= P[x_1 = A, x_2 = B] / P[x_2 = B] \\
 &= 0.5 / 0.58 \approx 0.8621
 \end{aligned}$$

$$P[x_1 = A | x_2 = B] > \gamma$$

$$\begin{aligned}
 IV. \quad P[x_2 = B | x_1 \neq A] &= P[x_1 \neq A, x_2 = B] / P[x_1 \neq A] \\
 &= 0.08 / (1 - 0.9) = 0.8
 \end{aligned}$$

$$P[x_2 = B | x_1 \neq A] > \alpha$$

Listing B-1 shows a MATLAB function that generates a dataset compliant with the constraints presented above.

```

function [z,Metadata] = example1(N)
% Example dataset for Appendix B
% Usage: [z,metaData] = example2(N)
% This dataset shows that if model merging is based on M, rather than Mh,
% convergence of the algorithm can be compromised. These results are for
% D, as using Dw and M leads to numerical ambiguity. (Divide by zero.)
% Input arguments:
% N      size of resultant dataset (N=5000 is a good value)
% The parameters that show these results are:
% theta = 89.99
% alpha = 0.1
% gamma = 0.85

```

Listing B-1 MATLAB routine for data generation

```

MetaData.names = {'x1', 'x2'};
MetaData.log = [0 0];
MetaData.int = [0 0];

p = [0.5 0.4 0.08 0.02];
A = 1;
B = 10;

ep = cumsum(p);
sp = [0 ep(1:length(ep)-1)];
bp = [sp' ep']*N;
bp(:,1) = bp(:,1)+1;
bp(:,3) = diff(bp')'+1;

z = randn(N,2);
z(bp(1,1):bp(1,2), :) = ones(bp(1,3),1)*[A B];
z(bp(2,1):bp(2,2),1) = A*ones(bp(2,3),1);
z(bp(3,1):bp(3,2),2) = B*ones(bp(3,3),1);

```

Listing B-1 (continued)

The main contamination detection algorithm was modified to perform model merging on the basis of $f(C, D, \gamma)$. Using a dataset generated with the routine listed above, the contamination detection algorithm was invoked. As the Listing B-2 shows, the routine does not converge; rather, the state of the models at iteration 2 is repeated at the subsequent iterations.

```

>> Debug.trace = 1; Debug.edf = 0; Debug.pause = 0;
>> [z, MetaData] = example1(5000);
>> M = process( z , 0.1 , 89.99 , 0.85 , MetaData , Debug );

Iteration 1

Hypothesized Models
[1]      (x1 == 1.000000)
[2]      (x2 == 10.000000)

Coverage Matrix
1.00000 0.55556
0.86207 1.00000

Results of Model Merge
1

Accepted Models
[1]      (x1 == 1.000000)

Iteration 2

Hypothesized Models
[1]      (x2 == 10.000000)

Coverage Matrix
1.00000 0.55556
0.86207 1.00000

Results of Model Merge
1

Accepted Models
[1]      (x1 == 1.000000)

```

Listing B-2 Model merging without convergence

```

Iteration 3
Hypothesized Models
[1]      (x2 == 10.000000)

Coverage Matrix
1.00000 0.55556
0.86207 1.00000

Results of Model Merge
1

Accepted Models
[1]      (x1 == 1.000000)

Iteration 4
Hypothesized Models
[1]      (x2 == 10.000000)

Coverage Matrix
1.00000 0.55556
0.86207 1.00000

Results of Model Merge
1

Accepted Models
[1]      (x1 == 1.000000)

```

Listing B-2 (continued)

B.3. Comparison of Merging Based on All versus Working Data

Section B.2 showed an instability resulting when the model merging process is based on all models and all data, i.e., $f(C, D, \gamma)$. This section compares two alternate cases, both involving only the hypothesized models of the current iteration, C^H . One case merges using all the data, i.e., $f(C^H, D, \gamma)$ and the other uses only the working data, i.e., $f(C^H, D^W, \gamma)$.

Since C^H consists of models extract solely from D^W , it might seem that using all the data is superfluous—the merging could use only D^W and obtain the same results as if D had been used. This conjecture ignores the effect of correlation between variables in C and C^H . This correlation can lead to different sets of contamination models on the basis of whether D^W or D is used in the model merging process.

For example, consider a dataset consisting of three variables, $\{x_1, x_2, x_3\}$. Assume the variables are distributed as independent unit normal variates, with the exception of the

following three contamination models: $C_1 \in \{x_1 = A_1\}$, $C_2 \in \{x_2 = A_2\}$, and $C_3 \in \{x_3 = A_3\}$.

If the seven conditions listed below are met, the final set of contamination models will differ:

- I. $C_1 \in \{x_1 = A_1\}$ is detected on the first iteration;
- II. $C_2 \in \{x_2 = A_2\}$ is not detected on the first iteration;
- III. $C_3 \in \{x_3 = A_3\}$ is not detected on the first iteration;
- IV. $C_2 \in \{x_2 = A_2\}$ is detected on the second iteration;
- V. $C_3 \in \{x_3 = A_3\}$ is detected on the second iteration;
- VI. Sufficient overlap exists between C_2 and C_3 , on the basis of D , to allow merging;
- VII. Insufficient overlap exists between C_2 and C_3 , based on D^w , to allow merging.

These seven conditions can be expressed by the following equations:

$$\begin{aligned}
I. \quad & P[x_1 = A_1] > \alpha \\
II. \quad & P[x_2 = A_2] < \alpha \\
III. \quad & P[x_3 = A_3] < \alpha \\
IV. \quad & P[x_2 = A_2 | x_1 \neq A_1] > \alpha \\
V. \quad & P[x_3 = A_3 | x_1 \neq A_1] > \alpha \\
VI. \quad & P[x_2 = A_2 | x_3 = A_3] > \gamma \\
VII. \quad & P[x_2 = A_2 | x_1 \neq A_1, x_3 = A_3] < \gamma
\end{aligned} \tag{B.7}$$

Assuming that $\gamma = 0.9$, $\alpha = 0.2$, and θ_c is sufficient to detect the contamination, the conditions of Equation B.7 are met by the a dataset with the following distribution:

$$\begin{aligned}
P[x_1 = A_1, x_2 = A_2, x_3 = A_3] &= 0.03 \\
P[x_1 = A_1, x_2 \neq A_2, x_3 \neq A_3] &= 0.22 \\
P[x_1 \neq A_1, x_2 = A_2, x_3 = A_3] &= 0.142 \\
P[x_1 \neq A_1, x_2 = A_2, x_3 \neq A_3] &= 0.018 \\
P[x_1 \neq A_1, x_2 \neq A_2, x_3 = A_3] &= 0.018 \\
P[x_1 \neq A_1, x_2 \neq A_2, x_3 \neq A_3] &= 0.572
\end{aligned} \tag{B.8}$$

The equations below verify that the conditions of Equation B.7 are met by the dataset described by Equation B.8.

$$\begin{aligned}
 I. \quad P[x_1 = A_1] &= P[x_1 = A_1, x_2 = A_2, x_3 = A_2] \\
 &\quad + P[x_1 = A_1, x_2 \neq A_2, x_3 \neq A_3] \\
 &= 0.03 + 0.22 \\
 &= 0.25 > \alpha
 \end{aligned}$$

$$\begin{aligned}
 II. \quad P[x_2 = A_2] &= P[x_1 = A_1, x_2 = A_2, x_3 = A_3] \\
 &\quad + P[x_1 \neq A_1, x_2 = A_2, x_3 = A_3] \\
 &\quad + P[x_1 \neq A_1, x_2 = A_2, x_3 \neq A_3] \\
 &= 0.03 + 0.142 + 0.018 \\
 &= 0.19 < \alpha
 \end{aligned}$$

$$\begin{aligned}
 III. \quad P[x_3 = A_3] &= P[x_1 = A_1, x_2 = A_2, x_3 = A_3] \\
 &\quad + P[x_1 \neq A_1, x_2 = A_2, x_3 = A_3] \\
 &\quad + P[x_1 \neq A_1, x_2 \neq A_2, x_3 = A_3] \\
 &= 0.03 + 0.142 + 0.018 \\
 &= 0.19 < \alpha
 \end{aligned}$$

$$\begin{aligned}
 IV. \quad P[x_2 = A_2 | x_1 \neq A_1] &= P[x_1 \neq A_1, x_2 = A_2] / P[x_1 \neq A_1] \\
 &= \frac{P[x_1 \neq A_1, x_2 = A_2, x_3 = A_3] + P[x_1 \neq A_1, x_2 = A_2, x_3 \neq A_3]}{1 - P[x_1 = A_1]} \\
 &= \frac{0.142 + 0.018}{1 - 0.25} \\
 &= 0.213 > \alpha
 \end{aligned}$$

$$\begin{aligned}
 V. \quad P[x_3 = A_3 | x_1 \neq A_1] &= P[x_1 \neq A_1, x_3 = A_3] / P[x_1 \neq A_1] \\
 &= \frac{P[x_1 \neq A_1, x_2 = A_2, x_3 = A_3] + P[x_1 \neq A_1, x_2 \neq A_2, x_3 = A_3]}{1 - P[x_1 = A_1]} \\
 &= \frac{0.142 + 0.018}{1 - 0.25} \\
 &= 0.213 > \alpha
 \end{aligned}$$

$$\begin{aligned}
 VI. \quad P[x_2 = A_2 | x_3 = A_3] &= P[x_2 = A_2, x_3 = A_3] / P[x_3 = A_3] \\
 &= \frac{P[x_1 = A_1, x_2 = A_2, x_3 = A_3] + P[x_1 \neq A_1, x_2 = A_2, x_3 = A_3]}{P[x_3 = A_3]} \\
 &= \frac{0.03 + 0.142}{0.19} \\
 &\cong 0.90526 > \gamma
 \end{aligned}$$

$$\begin{aligned}
\text{VII. } P[x_2 = A_2 | x_1 \neq A_1, x_3 = A_3] &= P[x_1 \neq A_1, x_2 = A_2, x_3 = A_3] / P[x_1 \neq A_1, x_3 = A_3] \\
&= \frac{P[x_1 \neq A_1, x_2 = A_2, x_3 = A_3]}{P[x_1 \neq A_1, x_2 = A_2, x_3 = A_3] + P[x_1 \neq A_1, x_2 \neq A_2, x_3 = A_3]} \\
&= \frac{0.142}{0.142 + 0.018} \\
&= 0.8875 < \gamma
\end{aligned}$$

Listing B-3 shows a MATLAB script that generates a dataset compliant with Equation B.8, where $A_1 = 11$, $A_2 = 12$, and $A_3 = 13$.

```

function [z,Metadata] = example2(N)
% Example dataset for Appendix B
% Usage: [z,metadata] = example2(N);
% This dataset shows that if model merging is based on D, rather than Dw,
% a more minimal set of contamination models will result. In either case,
% merging should be based on Mh, not M.
% The parameters that show these results are:
% theta = 89.9
% alpha = 0.2
% lambda = 0.9

Metadata.names = {'x1', 'x2', 'x3'};
Metadata.log = [0 0 0];
Metadata.int = [0 0 0];

p = [0.03 0.22 0.142 0.018 0.018 0.572];
ep = cumsum(p);
sp = [0 ep(1:length(ep)-1)];
bp = [sp' ep'];
bp = bp*N;
bp(:,1) = bp(:,1)+1;
bp(:,3) = diff(bp')'+1;
bp = round(bp);

A1 = 11;
A2 = 12;
A3 = 13;

z = randn(N,3);
z(bp(1,1):bp(1,2), :) = ones(bp(1,3),1)*[A1 A2 A3];
z(bp(2,1):bp(2,2),1) = A1*ones(bp(2,3),1);
z(bp(3,1):bp(3,2),2:3) = ones(bp(3,3),1)*[A2 A3];
z(bp(4,1):bp(4,2),2) = A2*ones(bp(4,3),1);
z(bp(5,1):bp(5,2),3) = A3*ones(bp(5,3),1);

```

Listing B-3 MATLAB data generation routine

When data generated from the function above is passed through the contaminate detection algorithm with merging based on $f(C^H, D, \gamma)$, two models result:

$C_1 \in \{x_1 = A_1\}$ and $C_2 \in \{x_2 = A_2\}$. This is verified by Listing B-4.

```

>> [D,MetaData] = example2(10000);
>> Debug.pause = 1; Debug.trace = 1; Debug.edf = 0;
>> m = process( D , 0.2 , 89.99 , 0.9 , MetaData , Debug );
Iteration 1

Hypothesized Models
  [1]      (x1 == 11.000000)

Coverage Matrix
  1.00000

Results of Model Merge
  1

Accepted Models
  [1]      (x1 == 11.000000)

Press a key to continue...

Iteration 2

Hypothesized Models
  [1]      (x2 == 12.000000)
  [2]      (x3 == 13.000000)

Coverage Matrix
  1.00000 0.90526
  0.90526 1.00000

Results of Model Merge
  1

Accepted Models
  [1]      (x1 == 11.000000)
  [2]      (x2 == 12.000000)

Press a key to continue...

```

Listing B-4 Merging results - two models retained

However, when the same data is passed through the algorithm where merging is based on only the working data, i.e., $f(C^H, D^W, \gamma)$, all three contamination models result. This is confirmed by Listing B-5.

```

>> [D,MetaData] = example2(10000);
>> Debug.pause = 1; Debug.trace = 1; Debug.edf = 0;
>> m = process( D , 0.2 , 89.99 , 0.9 , MetaData , Debug );
Iteration 1

Hypothesized Models
  [1]      (x1 == 11.000000)

Coverage Matrix
  1.00000

Results of Model Merge
  1

Accepted Models
  [1]      (x1 == 11.000000)

Press a key to continue...

Iteration 2

```

Listing B-5 Merging results – three models retained

```

Hypothesized Models
[1]      (x2 == 12.000000)
[2]      (x3 == 13.000000)

Coverage Matrix
1.00000 0.88750
0.88750 1.00000

Results of Model Merge
1      2

Accepted Models
[1]      (x1 == 11.000000)
[2]      (x2 == 12.000000)
[3]      (x3 == 13.000000)

Press a key to continue...

```

Listing B-5 (continued)

Comparing the results of merging based on $f(C^H, D, \gamma)$ and $f(C^H, D^W, \gamma)$, a tradeoff is revealed between the minimal number of models and “smoothness” of the final data. When the merging is based on $f(C^H, D, \gamma)$, a smaller number of models results; however, since there is some non-zero amount of unmodeled contamination, i.e., $P[x_3 = A_3 | x_1 \neq A_1, x_2 \neq A_1] \neq 0$, the final data will have a small amount of contamination remaining. This is illustrated in Figure B-1.

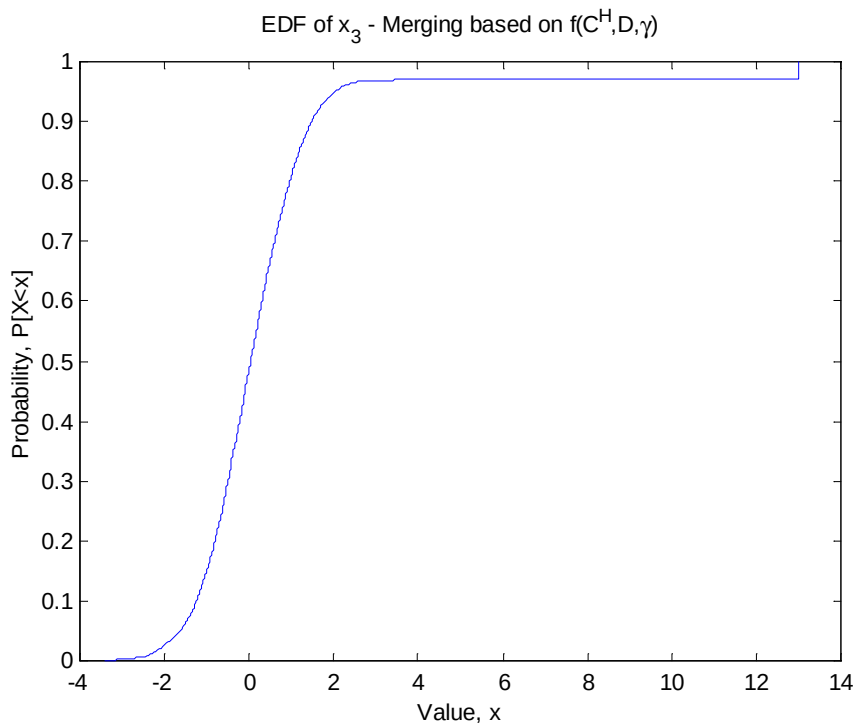


Figure B-1 Data after detection, merging based on all data

Alternatively, the merging can be based on $f(C^H, D^W, \gamma)$. This results in a non-minimal number of models, but a smoother final dataset, as illustrated by Figure B-2.

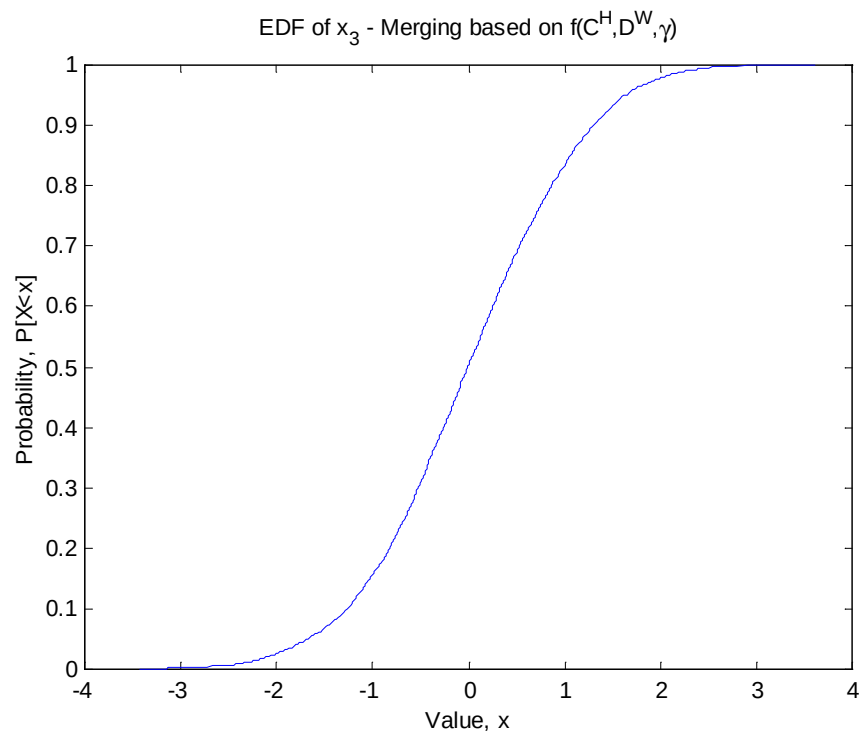


Figure B-2 Data after detection, merging based on working data

Appendix C: MATLAB Source for Contamination Detection Algorithm

This appendix contains the source code for the MATLAB implementation of the contamination detection algorithm developed in Chapter 4. The source listings are organized similarly to the organization of Section 4.3. First the overall algorithm is presented, then each sub-process.

C.1. Overall Algorithm

Listing C-1 shows the program process.m, which implements the contamination detection algorithm.

```
function [M,iter] = process( z , alpha , theta , gamma , MetaData , Debug )
% Process.m - Detect contamination regions using the algorithm described in
% Chapter 4.
% Usage M = process( D , alpha , theta , gamma , MetaData , Debug )
% D      the input dataset, expressed as a matrix with each row being
%         an independent observation
% alpha   the minimum amount of contamination to detect
% theta   the measure of the contamination severity
% gamma   the minimum overlap to use during model merging
% MetaData The model metadata. The elements are:
%   MetaData.names
% Debug    A structure of debug flags. The elements are (all are 0/1):
%   Debug.edf      Plot intermediate EDFs showing the contamination regions.
%   Debug.trace    Print verbose console information
%   Debug.pause    Await console input after each iteration

M = {};
iter = 1;
while iter < 5
    Mh = {};
    Dw = cleanData( z , M , MetaData );
    Rh = [];
    v = [];
    for i=1:size(z,2)
        Rv = contaminateRegions( Dw, i , alpha , theta , MetaData , Debug );
        v = [v i*ones(1,length(Rv))];
        Rh = [ Rh Rv ];
    end
    if length(Rh) == 0
        go = 0;
        break;
    end
    for j=1:length(Rh)
        fm = formalizeModel(Dw,Rh{j},MetaData,v(j));
        Mh = [Mh fm];
    end

    cm = coverageMatrix( Dw , Mh , MetaData );
    i = modelMerge( cm , gamma );
    M = [ M Mh(i) ];
end
```

Listing C-1 process.m

```

if Debug.trace
    fprintf( 'Iteration %d\n\n' , iter );
    fprintf('\nHypothesized Models\n');
    printModels( Mh );
    fprintf('\nCoverage Matrix\n' );
    printMatrix( cm );
    fprintf('\nResults of Model Merge\n' );
    fprintf('\t%d',i);
    fprintf('\n');
    fprintf('\nAccepted Models\n');
    printModels( M );
    fprintf('\n\n' );
end
if Debug.pause
    fprintf('Press a key to continue...\n\n'); pause;
end

iter = iter + 1;
end

function printMatrix( x )
for i=1:size(x,1)
    fprintf('\t%1.5f',x(i,:));
    fprintf('\n');
end
end

```

Listing C-1 (continued)

C.2. Data Censoring

Listing C-2 shows the program cleanData.m, which implements the censoring of data on the basis of prior accepted models, as per Section 4.3.1.

```

function z = cleanData( x , M , MD )
%cleanData Returns data without contaminate-model associated elements.

for i=1:length(MD.names)
    cmd = sprintf( '%s = x(:,%d);' , MD.names{i} , i );
    eval(cmd);
end

bi = [];
for i=1:length(M)
    mi = find( eval( M(i).ruleString ) );
    bi = union( mi , bi );
end
gi = setdiff( 1:size(x,1) , bi );
z = x( gi , : );

```

Listing C-2 cleanData.m

C.3. Single-Variate Contamination Detection

Listing C-3 shows the program contaminateRegions.m that implements the sliding window process described in Section 4.3.2. Listing C-4 shows the function contiguousRegions.m which is used to determine contiguous regions of indices.

```

function R = contaminateRegions( Data , vi , alpha , theta , MetaData , Debug )
% contaminateRegions Finds regions of contamination from an EDF
% Given a data set and detection parameters, contaminateRegions
% returns the indices of points within regions meeting the
% detection criteria.
%
% Usage:
% R = contaminateRegions( Data , vi , alpha , theta , MetaData , Debug )
% R Regions of contamination. A array of indices.
% vi The variable index of data to be detected.
% theta The critical angle. A measure of contamination "clustering".
% alpha The minimum contamination level. A measure of contamination
% severity.
% MetaData Information structure about data being modeled.
% Debug A structure of flags used for debugging. Applicable flags are:
% Debug.edf {0,1} toggles edf figure display
%
% See Also: contiguousRegions.m

if( nargin == 5 )
    Debug.edf = 0;
end
R = [];
x = Data(:,vi)'; % initialize output
if( MetaData.log(vi) ) % extract the variable data
    x = log10(x); % convert to log scale if required
end

ca = theta*pi/180.0; % convert to radians
wl = ceil(alpha*length(x)); % calculate minimum window length

[xo,io] = sort(x ); % sort the data; preserve indices
y = (1:length(xo))./length(xo) ; % edf ordinate
xn = (xo - min(xo))./(max(xo)-min(xo)); % normalize edf abscissa

%warns = warning('off'); % turn off warnings (Inf. slope)
warning off;
i1 = 1:(length(xo)-wl); % starting indices
i2 = i1 + wl; % stopping indices
slp = (y(i2)-y(i1))./(xn(i2)-xn(i1)); % slopes regions starting at i1

cr = find(atan(slp)>ca); % initial contamination regions
[start,stop] = contiguousRegions(cr); % breakpoints of regions

id = [];
for j=1:length(start) % for all regions
    rb = start(j); % region begining point
    re = max([stop(j) rb+wl]); % region ending point
    rs = (y(re)-y(rb))/(xn(re)-xn(rb)); % calculate region slope
    while( atan(rs)>ca & re < length(xn) ) % grow the region until slope not
met
        re = re + 1;
        rs = (y(re)-y(rb))/(xn(re)-xn(rb));
    end
    if atan(rs)<ca % shrink region by 1, to meet slope
        re = re - 1;
    end
    R{j} = io(rb:re); % save the indices of the region
    id = [id rb:re]; % save the indices for debug plot
end
%warning(warns); % set warnings the way they were

if Debug.edf
    figure;
    plot(xo,y,xo(id),y(id),'r.');
    title( sprintf('EDF of %s',MetaData.names{vi}));
    xlabel('Value, x');
    ylabel('Probability, P[X<=x]');
end

```

Listing C-3 contaminateRegions.m

```

function [start,stop] = contiguousRegions( i )
% contiguousRegions Finds contiguous regions of indices
% [start,stop] = contiguousRegions(i) finds all blocks of
% consecutively numbered elements in i. start and stop are
% inclusive. Blocks of size 1 are reported.
%
% For example contiguousRegions( [1 2 3 4 8 9 10 12] ) returns
% start = [ 1 8 12 ] and
% stop = [ 4 10 12 ].
%
% See also: contaminateRegions.m

if length(i) == 0
    start = [];
    stop = [];
else
    start_index = [0 find(diff(i)>1)]+1;
    n = length(start_index);
    stop_index = [ start_index(2:n)-1 length(i)];
    start = i(start_index);
    stop = i(stop_index);
end

```

Listing C-4 contiguousRegions.m

Listing C-5 shows the program formalizeModel.m. This program accomplishes the model formalization described in Section 4.3.2.1.

```

function M = formalizeModel( x , ri , MD , vi )
%formalizeModel Returns a rule-based string denotation of a contamination
region.
% Model = formalizeModel( Data , RegionIndices , MetaData , VariableIndex )
% Data A matrix of observations.
% RegionIndices The rows of Data that correspond to the model
% MetaData Information about the data.
% VariableIndex The column of Data corresponding to the variable to be
modeled.

md = x(ri,vi); % model data
vd = x(:,vi); % variable data

mmax = max(md); % model maximum value
mmin = min(md); % model minimum value
vmax = max(vd); % variable maximum value
vmin = min(vd); % variable minimum value

name = MD.names{vi};
if ( MD.log(vi) & ~MD.int(vi) )
    name = sprintf( 'log10(%s)' , name);
    mmax = log10( mmax );
    mmin = log10( mmin );
    vmax = log10( vmax );
    vmin = log10( vmin );
end

if ( mmax == mmin )
    M.ruleString = sprintf( '(%s == %f)' , name , mmax );
elseif ( mmax == vmax )
    M.ruleString = sprintf( '(%s >= %f)' , name , mmin );
elseif ( mmin == vmin )
    M.ruleString = sprintf( '(%s <= %f)' , name , mmax );
else
    M.ruleString = sprintf( '(%f <= %s) & (%s <= %f)' , ...
        mmin , name , name , mmax );
end

```

Listing C-5 formalizeModel.m

C.4. Model Merging

Listing C-6 shows the function `coverageMatrix.m`, used to calculate the coverage matrix. Listing C-7 shows the program `modelMerge.m`, which merges the models as described in Section 4.3.3.

```
function cm = coverageMatrix( z , M , MD )
%coverageMatrix    Generates the coverage matrix using data, models, and meta
data.
%    cm = coverageMatrix( z , M , MD )

for i=1:length(MD.names)
    cmd = sprintf( '%s = z(:,%d);' , MD.names{i} , i );
    eval(cmd);
end

cm = eye( length( M ) );
for row = 1:length(cm)
    for col = 1:length(cm)
        a = find( eval( M(row).ruleString ) );
        b = find( eval( M(col).ruleString ) );
        cm(row,col) = length(find(ismember( a , b )))/length(a);
    end
end
```

Listing C-6 coverageMatrix.m

```
function i = modelMerge( cm , l )
%modelMerge        Merges models based on the coverage matrix and a minimum level.

aug = [(1:length(cm))', cm];

n = size(aug,2);
[v,r] = max( aug(:,2:n) - eye(n-1) );
[v,c] = max(v);
r = r(c);

while( v > l )
    aug(r,:) = [];
    aug(:,r+1) = [];
    n = size(aug,2);
    [v,r] = max( aug(:,2:n) - eye(n-1) );
    [v,c] = max(v);
    r = r(c);
end
i = aug(:,1);
```

Listing C-7 modelMerge.m

Appendix D: Scale Invariance of Distance Ratio

The formulation of the estimation of scale using the ratio of relative distances presented in Section 5.3.4.2 relies on the scale invariance of the distance ratio, $\frac{d_{\mu+\sigma}}{d_\mu}$. This appendix presents a proof of that assertion.

Consider the unit normal variate. The distances are defined by:

$$d_\mu = \Phi^{-1}\left(\frac{1}{2}(1+\alpha)\right) - \Phi^{-1}\left(\frac{1}{2}(1-\alpha)\right) = 2\Phi^{-1}\left(\frac{1}{2}(1+\alpha)\right)$$

$$d_{\mu+\sigma} = b - a \quad \frac{a+b}{2} = \mu + \sigma \wedge \Phi(b) - \Phi(a) = \alpha$$

The distance ratio is given by:

$$\frac{d_{\mu+\sigma}}{d_\mu} = \frac{b-a}{2\Phi^{-1}\left(\frac{1}{2}(1+\alpha)\right)}$$

For an arbitrary normal distribution, with parameters μ and σ , the standardization theorem [19] gives the new values of the endpoints of each region:

$$d_{\mu+\sigma} \rightarrow b' = \mu + \sigma b \quad a' = \mu + \sigma a$$

$$d_\mu \rightarrow \mu + \sigma\Phi^{-1}\left(\frac{1}{2}(1+\alpha)\right) - \mu + \sigma\Phi^{-1}\left(\frac{1}{2}(1-\alpha)\right)$$

The distance ratio becomes:

$$\begin{aligned} \frac{d_{\mu+\sigma}}{d_\mu} &= \frac{\mu + \sigma b - (\mu + \sigma a)}{\mu + \sigma\Phi^{-1}\left(\frac{1}{2}(1+\alpha)\right) - (\mu + \sigma\Phi^{-1}\left(\frac{1}{2}(1-\alpha)\right))} \\ &= \frac{\sigma(b-a)}{\sigma(\Phi^{-1}\left(\frac{1}{2}(1+\alpha)\right) - \Phi^{-1}\left(\frac{1}{2}(1-\alpha)\right))} \\ &= \frac{b-a}{\Phi^{-1}\left(\frac{1}{2}(1+\alpha)\right) - \Phi^{-1}\left(\frac{1}{2}(1-\alpha)\right)} \\ &= \frac{b-a}{2\Phi^{-1}\left(\frac{1}{2}(1+\alpha)\right)} = \frac{d_{\mu+\sigma}}{d_\mu} \end{aligned}$$

Since the distance ratio for an arbitrary normal distribution is the same as that of the unit normal variate, the distance ratio is scale invariant.

Appendix E: MATLAB Source for Multimodal Estimation Techniques

This appendix contains the source code for the MATLAB implementation of the various estimation techniques presented in Chapter 5. Additionally, the code used to generate data from a Gaussian mixture is presented.

E.1. Gaussian Mixture Generation

Listing E-1 shows the program multimodal.m used to generate observations from a Gaussian mixture.

```
function [x,i]=multimodal(w,m,s,n);
%Generates multimodal Guassian mixtures
%
% [x,i] = multimodal(w,m,s,n)
% x = generated data
% i = true underlying distribution
% w = vector of wieghts
% m = vector of means
% s = vector of standard deviations
% n = number of observations to generate

w = w./sum(w);
[p,a] = aliastable(w);
i = aliasgen(p,a,n);
x = m(i)+s(i).*randn(1,length(i));
```

Listing E-1 multimodal.m

The program relies on two addition programs, aliasgen.m and aliastable.m. These are used to generate random variates from a finite set with arbitrary probability. Listing E-2 and Listing E-3 show these programs.

```
function x = aliasgen( p , a , n );
% Generates random numbers from a finite set with arbitrary probabilities.
%
% [x] = aliasgen( p , a , n )
% p = probability of keeping the the value
% a = alias, if the value is rejected
% n = number of observations to generate
%
% See also: aliastable

u = rand(1,n);
b = floor(u*length(p))+1;
pa = length(p)*u - b + 1;
x = (pa<p(b)).*b + (pa>p(b)).*a(b);
```

Listing E-2 aliasgen.m

```

function [p,a] = aliastable( w );
% Generates a probability and alias tables for random number generation.
% [p,a] = aliastable( w )
% p = probability of keeping the the value
% a = alias, if the value is rejected
% w = probailities of occurence
%
% See also: aliasgen

a = zeros(size(w));
p = w./length(w);
[valmax,posmax] = max(p);
while( abs(valmax-1.0) > 0.00001 )
    set = find(a~=0);
    pt = p;
    pt(set) = Inf;
    [valmin,posmin] = min(pt);
    p(posmax) = p(posmax) - (1-valmin);
    a(posmin) = posmax;
    [valmax,posmax] = max(p);
end

```

Listing E-3 aliastable.m

E.2. Method of Moments

Listing E-4 shows the program gmmom.m which implements method of moments estimation.

```

function p = gmmom(z)
% Parameter estimation of a 2-component Gaussian mixture using method of
moments.
% P = GMMOM(Z)
% Z = data
% P = estimate of parameters [w1 m1 s1 w2 m2 s2]

p = [];
emoment = [];
% Generate the central moments
u1 = mean(z);
u2c = mean( (z-u1).^2 );
u3c = mean( (z-u1).^3 );
u4c = mean( (z-u1).^4 );
u5c = mean( (z-u1).^5 );
% Generate the non-central moments
u2 = mean( z.^2 );
u3 = mean( z.^3 );
u4 = mean( z.^4 );
u5 = mean( z.^5 );

% Solve for the intermediate variables phi and phi'
k4 = u4c - 3*u2c^2;
k5 = u5c - 10*u2c*u3c;
a0 = -24*u3c^6;
a1 = -96*u3c^4*k4;
a2 = -63*u3c^2*k4^2 - 72*u3c^3*k5;
a3 = 288*u3c^4 - 108*u3c*k4*k5 + 27*k4^3;
a4 = 444*u3c^2*k4 - 18*k5^2;
a5 = 90*k4^2 + 72*u3c*k5;
a6 = 36*u3c^2;
a7 = 84*k4;
a8 = 0;
a9 = 24;
a = [a9 a8 a7 a6 a5 a4 a3 a2 a1 a0];

```

Listing E-4 gmmom.m

```

r = roots(a);
r = r(find(imag(r)==0));
r = r(find(r<0));
% r is now the set of candidate values for phi
% we will iterate over all the possible phi's (and phi')
% to find the set that gives the smallest error.
% that one will be our estimate
for i = 1:length(r)
    phi = r(i);
    c = [2*phi^2 4*phi*u3c (-6*phi^3 - 3*k4*phi - u3c^2)];
    phi_prime = roots(c);
    for j=length( phi_prime )
        theta = sort( roots([ 1 -phi_prime(j) phi ]) );
        m1 = theta(1) + u1;
        m2 = theta(2) + u1;
        w1 = theta(2)/(theta(2) - theta(1));
        w2 = 1 - w1;
        d = [0;0];
        d(1) = u2 - w1*m1^2 - w2*m2^2;
        d(2) = u3 - w1*m1^3 - w2*m2^3;
        b = [w1 w2; 3*w1*m1 3*w2*m2];
        s = sqrt(inv(b)*d);
        params = [w1 m1 s(1) w2 m2 s(2)];
        if( (imag(s(1)) == 0 ) & (imag(s(2)) == 0) )
            % calculate the analytic moments based on these parameters
            om = [u1 u2 u3 u4 u5];
            am = gmmoments( [w1 m1 s(1) w2 m2 s(2)] );
            em = max(abs( (am - om)./om ));

            p = [p ; params];
            emoment = [emoment; em];
        end
    end
end
[0,i] = sort(emoment);
p = p(i,:);
p = p(1,:);

```

Listing E-4 (continued)

E.3. Maximum Likelihood Estimation

Listing E-5 shows the program gmmle.m used to estimate the parameters of a Gaussian mixture using maximum likelihood estimation.

```

function [p,c] = gmmle(z,guess)
% Maximum likelihood parameters for a 2 component Guassian mixture.
% [P,C] = GMMLE(guess , Z )
% guess = initial guess, of form [w1 m1 s1 w2 m2 s2]
% Z = data
% Returns
% P = MLE parameters, in same form as guess
% C = 1 if convergence, 0 if convergence failed
% See also: gmmleobj.m
pi = [guess(1) guess(2) guess(3) guess(5) guess(6)];
[x,it] = fmins('gmmleobj',pi,[],[],z);
%options = optimset;
%options.MaxIter = 100;
%[x,it] = fminsearch('gmmleobj',pi,[],z);
if( it(10) == it(14) )
    c = 0;
else
    c = 1;
end
p = [x(1) x(2) x(3) 1-x(1) x(4) x(5)];

```

Listing E-5 gmmle.m

The program used the function `gmmleobj.c` as the object function to be minimized. This function is the log-likelihood function shown in Listing E-6.

```
function j = gmmleobj(x,z)
% The Gaussian mixture (2 component) maximum-likelihood objective function.
% J = GMMLEOBJ(X,Z)
% X = the value of the parameters, x = [w1 m1 s1 m2 s2]
% Z = the dataset
% Returns the value of the objective function.
w1 = x(1);
m1 = x(2);
s1 = x(3);
w2 = 1-x(1);
m2 = x(4);
s2 = x(5);
j = w1*normpdf(z,m1,s1) + w2*normpdf(z,m2,s2);
j = -sum(log(j));
```

Listing E-6 gmmleobj.m

E.4. Midpoint Distance Profile

The program `mpdpest.m` of Listing E-7 implements the midpoint distance profile estimation technique.

```
function [y,mp,d] = mpdpest( x , alpha , debug )
% Y = MPDPEST( X , ALPHA , debug )
% Estimate the parameters of a Gaussian mixture using the midpoint
% distance profile method.

if( nargin < 3 )
    debug.plots = 0;
    debug.trace = 0;
end

[mp,d] = mpdprofile( x , alpha );

% Use a polynomial approximation to locate the extrema.
po = 9;
p = polyfit(mp,d,po);
dp = [(po):-1:1].*p(1:po);
ddp = [(po-1):-1:1].*dp(1:(po-1));
extrema = roots(ddp);
extrema = extrema(find(extrema>min(mp)));
extrema = extrema(find(extrema<max(mp)));
extrema = extrema(find(imag(extrema)==0));
extrema = sort(extrema);

% Ensure that the extrema alternate minima, maxima, etc.
% If they don't, issue a warning.
concavity = sign(polyval(ddp,extrema));
start = min(find(concavity==1));
stop = max(find(concavity==1));
extrema = extrema(start:stop);
concavity = concavity(start:stop);
n = ceil( length(extrema) / 2 );
t = [ones(1,n); -ones(1,n)];
t = t(1:size(extrema));
if( (t*concavity) ~= size(extrema) )
    warning('Extrema of the profile not standard min-max alternation.');
```

Listing E-7 mpdpest.m

```

inflection = roots(ddp);
inflection = inflection(find(imag(inflection)==0));
inflection = inflection(find(inflection>min(extrema)));
inflection = inflection(find(inflection<max(extrema)));
inflection = sort(inflection);

if( debug.trace )
    fprintf( 'Coefficients of the polynomial approximation:\n\t%s\n\n' , ...
        sprintf( '%5.4f ' , p ) );
    fprintf( 'Location of extrema:\n\t%s\n\n' , ...
        sprintf( '%5.4f ' , extrema ) );
    fprintf( 'Location of inflection points\n\t%s\n\n' , ...
        sprintf( '%5.4f ' , inflection ) );
end

if( debug.plots )
    figure;
    plot(mp,d,'r.',mp,polyval(p,mp),'k');
    xlabel( 'Midpoint, mp_i' );
    ylabel( 'Distance, d_i' );
    legend( 'Actual' , 'Polynomial Approx.' );
end

% Find the regions of "influence" for the minima.
minima_ind = find( concavity==1 );
nreg = length(minima_ind);
regions = zeros(nreg,3);
reg = zeros(nreg,2);
regions(:,1) = extrema(minima_ind);
for i = 1:nreg
    c = regions(i,1);
    lb = max(inflection(find(inflection<c)));
    ub = min(inflection(find(inflection>c)));
    if( length(lb) == 0 )
        lb = -Inf;
    end
    if( length(ub) == 0 )
        ub = Inf;
    end
    regions(i,2:3) = [lb ub];
    regs(i,1) = min(find( mp > lb ));
    regs(i,2) = max(find( mp < ub ));
end

if( debug.trace )
    fprintf( 'Regions of Influence of Minima\n' );
    fprintf( '\tMinima\tStart\tEnd\n' );
    for i=1:length(regions(:,1))
        fprintf( '\t%5.4f\t%5.4f\t%5.4f\n' , regions(i,:) );
    end
    fprintf( '\n' );
end

bc = zeros(nreg,3);
for i=1:nreg
    k = regs(i,1):regs(i,2);
    p = polyfit( mp(k) , d(k) , 2 );
    bc(i,:) = p;
end

% remove any polynomials with negative a coeffs and issue warning
bi = find(bc(:,1)<0);
if(length(bi)>0)
    warning( 'Poor binomial fits to some regions, regions dropped.' );
    bc = bc(find(bc(:,1)>0),:)
end

```

Listing E-7 (continued)

```

if( debug.trace )
    fprintf( 'Binomial Coefficients\n' )
    for i=1:length(bc(:,1))
        fprintf( '\t%5.4f\t%5.4f\t%5.4f\n' , bc(i,:) );
    end
    fprintf( '\n' );
end

if( debug.plots )
    figure;
    plot( mp,d );
    hold on
    for i=1:nreg
        k = regs(i,1):regs(i,2);
        plot(mp(k),d(k), 'rx');
    end
    hold off
    xlabel( 'Midpoint, mp_i' );
    ylabel( 'Distance, d_i' );
    legend( 'All Data' , 'R_i' );

    figure;
    plot( mp,d, ':' );
    hold on
    for i=1:nreg
        k = regs(i,1):regs(i,2);
        plot(mp(k),polyval(bc(i,:),mp(k)));
    end
    hold off;
    xlabel( 'Midpoint, mp_i' );
    ylabel( 'Distance, d_i' );
    legend( 'Profile' , 'Binomial Approximations' );
end

locs = -bc(:,2)./bc(:,1)./2;
cf = 1.656 - 0.359*alpha;
for i=1:length(locs)
    dist(i,1) = polyval(bc(i,:),locs(i));
    scale(i,1) = sqrt( dist(i).*(cf-1)./bc(i,1) );
end
w = alpha./(2*normcdf(dist./2./scale)-1);
w = w./sum(w);
y = [w, locs, scale];

```

Listing E-7 (continued)

The program mpdprofile.m generates the midpoint distance profile using a discrete filtering approach. This program is shown in Listing E-8.

```

function [mp,d] = mpdprofile( z , alpha );

z = sort(z);
v = floor( alpha*length(z) ) + 1;
d = filter( [ 1 zeros(1,v-2) -1 ] , 1 , z );
mp = filter( [0.5 zeros(1,v-2) 0.5] , 1 , z );
d = d(v:length(d));
mp = mp(v:length(mp));

```

Listing E-8 mpdprofile.m

Appendix F: Multimodal Estimation Simulation Results

This appendix discusses simulation experiment and results used to assess the performance of the three estimation techniques presented in Chapter 5. First, the simulation experiment is described. Then the results for bias, rate of convergence and relative efficiency are discussed.

F.1. Simulation Experiment

The three properties studied require multiple observations of the estimator random variable. Therefore, the experiment involved generating m samples from two-component Gaussian mixture with $\{w_1, \mu_1, \sigma_1, w_2, \mu_2, \sigma_2\} = \{0.4, 0, 2, 0.6, 8, 3\}$ and estimating the parameters using the three techniques.

This experimental trial was repeated several times to obtain multiple observations of the estimator. For $m \in \{2500, 5000, 7500, 10000, 12500, 15000, 20000\}$, 500 trials, at each value of m , were performed and 50 trials were conducted for each value of $m \in \{25000, 30000, 35000, 40000, 45000, 50000\}$. These observations were used to evaluate the each estimator's bias, rate of convergence, and efficiency.

F.2. Estimator Bias

Two measures of bias were evaluated. The first value compared the estimates of the parameters to the actual values of the parameters, as per Equation F.1.

$$bias = |\hat{\theta} - \theta| \quad (F.1)$$

Since the observations of the data were finite, the empirical parameters will differ from the actual parameters of the underlying dataset. For example, if 100 samples are drawn, perhaps only 39 will come from the first component, so, for that dataset, $w_1 = 0.39$ rather than the “true” value of $w_1 = 0.40$. To correct for this difference, inherent in observing a finite sample, the bias was also calculated against the observational values of the parameters, as per Equation F.2

$$bias = |\hat{\theta} - \theta_{observation}| \quad (F.2)$$

Figures F-1 through F-6 show the average actual, or “true”, bias and the average observational bias for the three estimation techniques as a function of the sample size. The bias for w_1 and w_2 are equivalent, so the plots overlap. Section 5.5.1 discusses and interprets these figures.

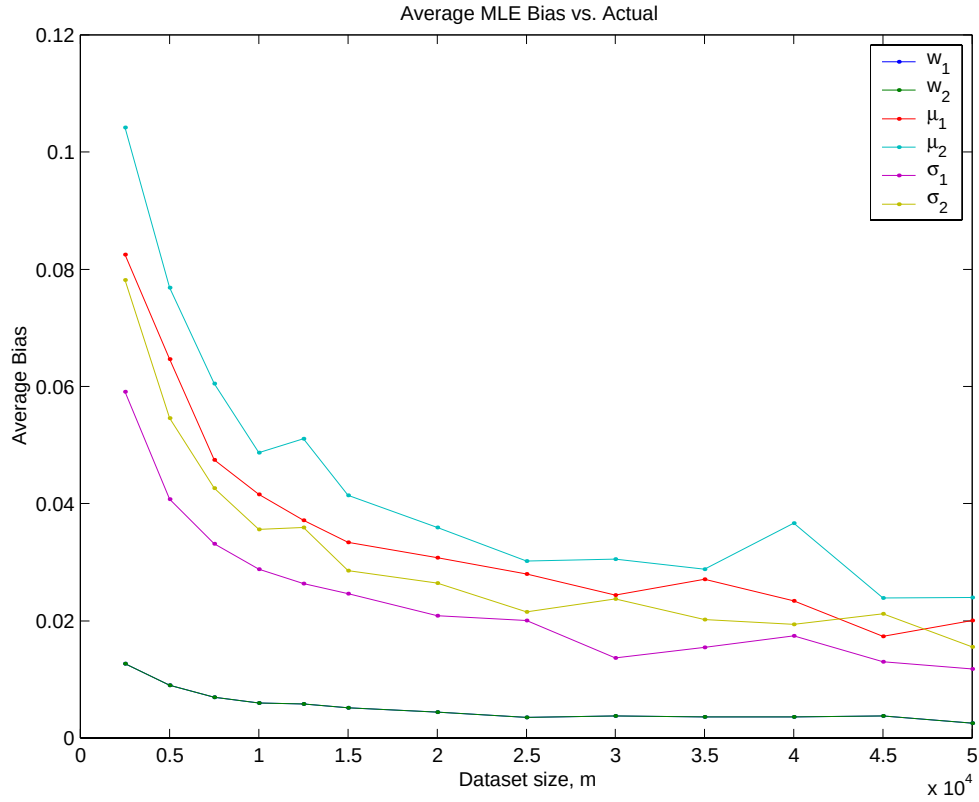


Figure F-1 Average Actual Bias (MLE)

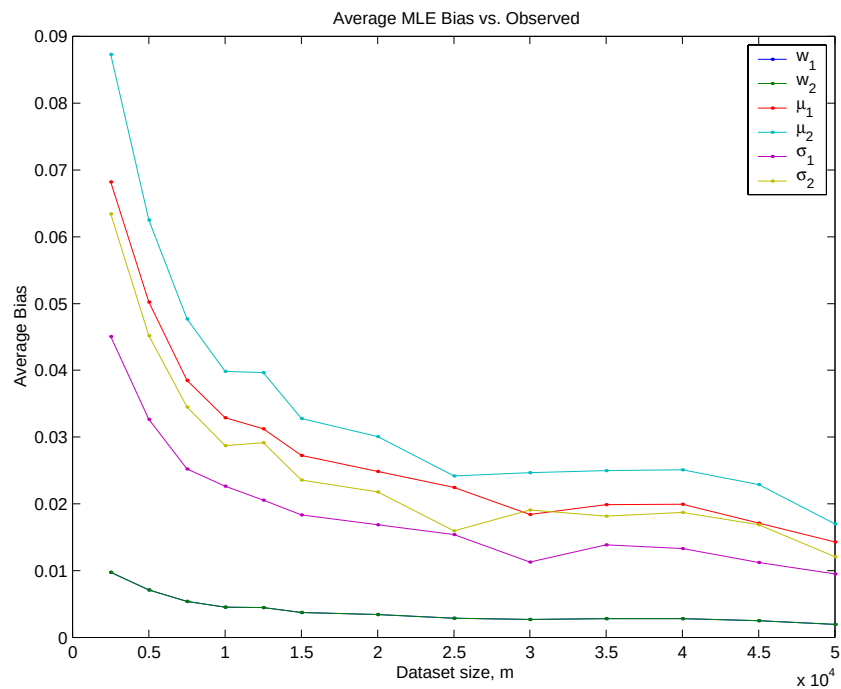


Figure F-2 Average Observation Bias (MLE)

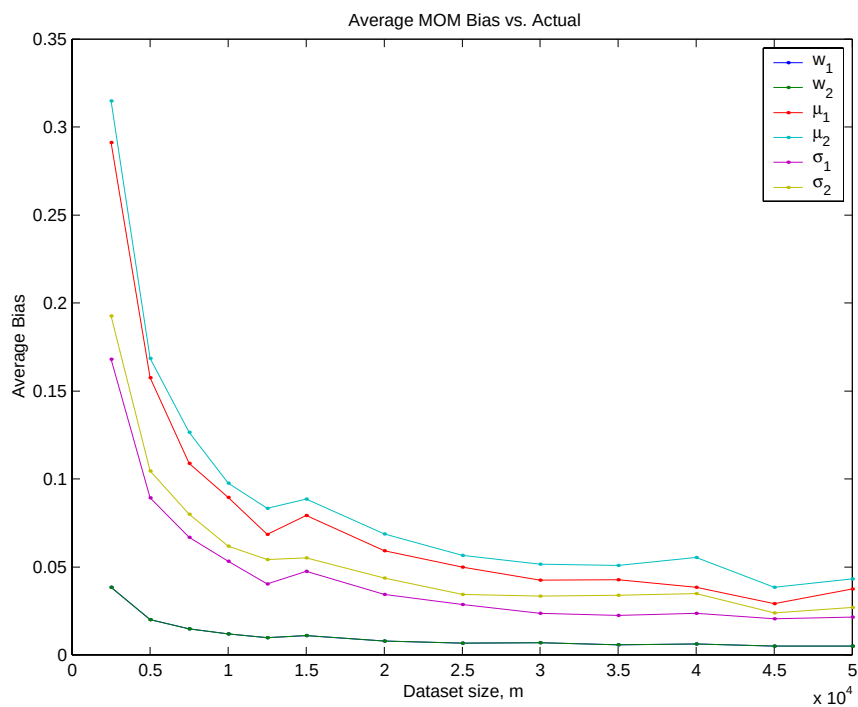


Figure F-3 Average Actual Bias (MOM)

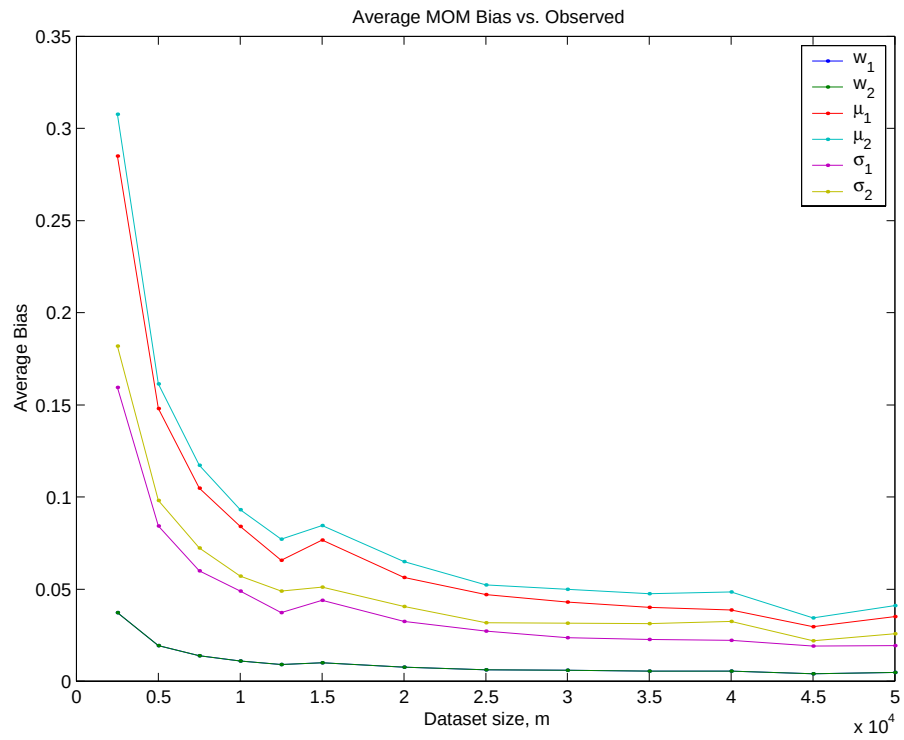


Figure F-4 Average Observation Bias (MOM)

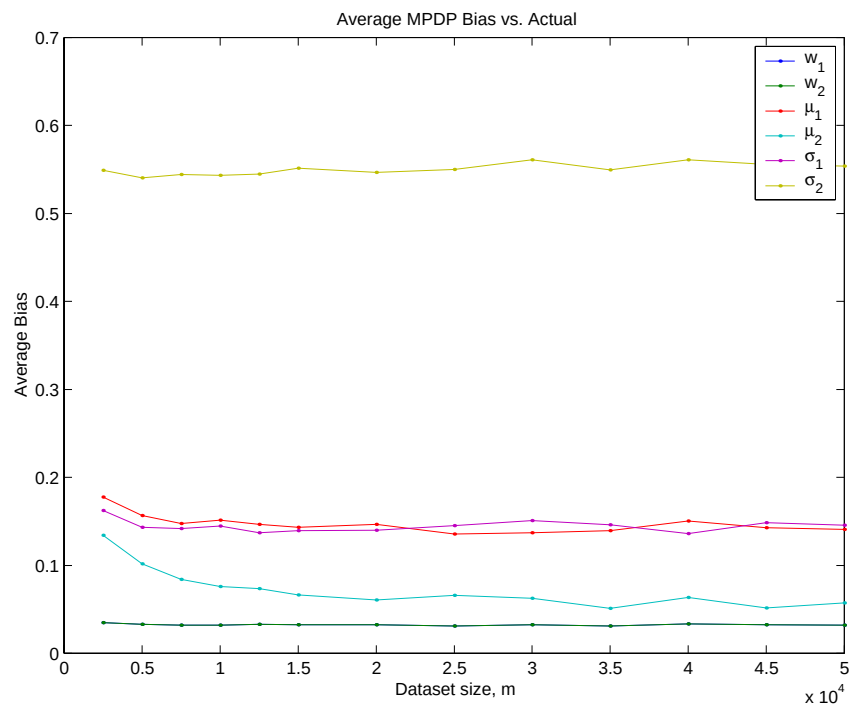


Figure F-5 Average Actual Bias (MPDP)

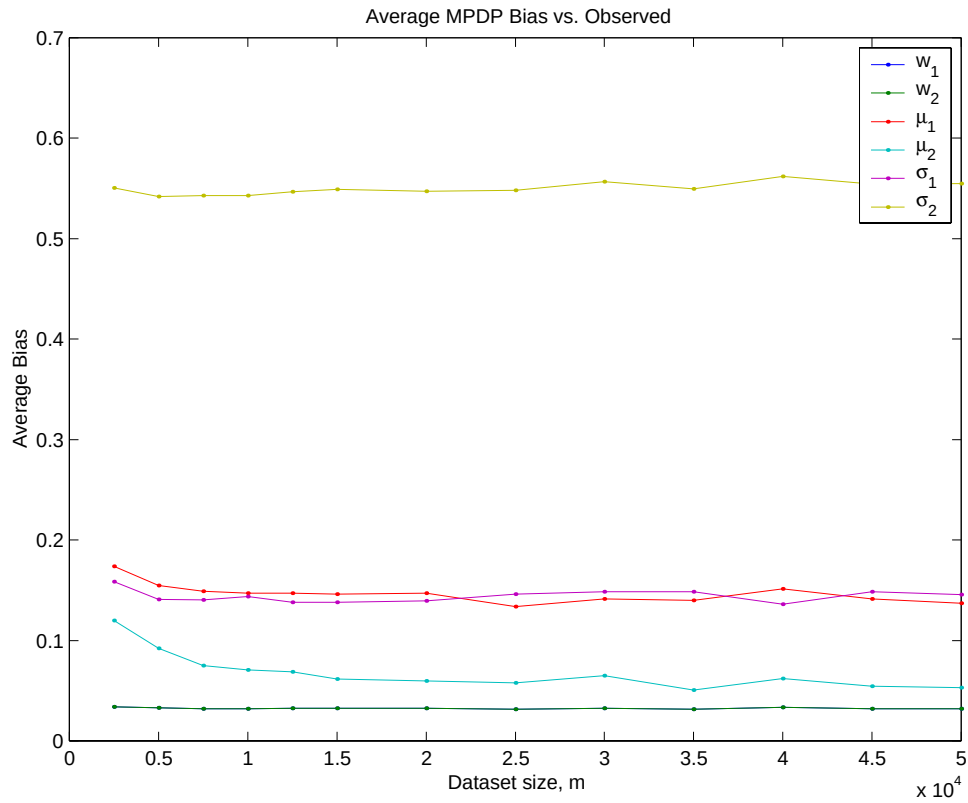


Figure F-6 Average Observation Bias (MPDP)

Tables F-1 through F-6 present the same information in tabular form.

Table F-1 Actual MLE Bias

	w1	w2	m1	m2	s1	s2
n=2500	0.01267	0.01267	0.08252	0.10423	0.05909	0.07814
n=5000	0.00897	0.00897	0.06466	0.07687	0.04078	0.05458
n=7500	0.00698	0.00698	0.04750	0.06048	0.03315	0.04261
n=10000	0.00596	0.00596	0.04159	0.04867	0.02882	0.03561
n=12500	0.00582	0.00582	0.03719	0.05105	0.02637	0.03593
n=15000	0.00516	0.00516	0.03343	0.04139	0.02462	0.02855
n=20000	0.00442	0.00442	0.03074	0.03594	0.02088	0.02641
n=25000	0.00350	0.00350	0.02798	0.03017	0.02003	0.02156
n=30000	0.00377	0.00377	0.02436	0.03051	0.01366	0.02377
n=35000	0.00356	0.00356	0.02712	0.02879	0.01550	0.02022
n=40000	0.00359	0.00359	0.02341	0.03664	0.01740	0.01943
n=45000	0.00376	0.00376	0.01737	0.02393	0.01304	0.02117
n=50000	0.00256	0.00256	0.02005	0.02395	0.01181	0.01558
Average	0.00544	0.00544	0.03676	0.04559	0.02501	0.03258

Table F-2 Observation MLE Bias

	w1	w2	m1	m2	s1	s2
n=2500	0.00978	0.00978	0.06818	0.08727	0.04505	0.06342
n=5000	0.00710	0.00710	0.05021	0.06250	0.03264	0.04517
n=7500	0.00540	0.00540	0.03852	0.04770	0.02524	0.03448
n=10000	0.00453	0.00453	0.03292	0.03986	0.02268	0.02875
n=12500	0.00445	0.00445	0.03122	0.03963	0.02055	0.02915
n=15000	0.00375	0.00375	0.02727	0.03276	0.01838	0.02356
n=20000	0.00343	0.00343	0.02487	0.03010	0.01686	0.02180
n=25000	0.00289	0.00289	0.02249	0.02418	0.01542	0.01595
n=30000	0.00272	0.00272	0.01844	0.02469	0.01127	0.01909
n=35000	0.00283	0.00283	0.01988	0.02500	0.01388	0.01817
n=40000	0.00283	0.00283	0.01995	0.02508	0.01331	0.01870
n=45000	0.00251	0.00251	0.01716	0.02293	0.01125	0.01691
n=50000	0.00196	0.00196	0.01432	0.01698	0.00954	0.01208
Average	0.00417	0.00417	0.02965	0.03682	0.01970	0.02671

Table F-3 Actual MOM Bias

	w1	w2	m1	m2	s1	s2
n=2500	0.03846	0.03846	0.29129	0.31501	0.16808	0.19278
n=5000	0.02000	0.02000	0.15764	0.16862	0.08934	0.10460
n=7500	0.01481	0.01481	0.10892	0.12650	0.06683	0.08009
n=10000	0.01191	0.01191	0.08955	0.09757	0.05317	0.06195
n=12500	0.00975	0.00975	0.06841	0.08336	0.04024	0.05431
n=15000	0.01091	0.01091	0.07938	0.08848	0.04755	0.05505
n=20000	0.00793	0.00793	0.05910	0.06884	0.03448	0.04357
n=25000	0.00676	0.00676	0.04984	0.05653	0.02874	0.03435
n=30000	0.00696	0.00696	0.04253	0.05160	0.02365	0.03347
n=35000	0.00578	0.00578	0.04282	0.05089	0.02233	0.03391
n=40000	0.00624	0.00624	0.03855	0.05530	0.02353	0.03484
n=45000	0.00496	0.00496	0.02911	0.03855	0.02052	0.02393
n=50000	0.00503	0.00503	0.03744	0.04326	0.02139	0.02693
Average	0.01150	0.01150	0.08420	0.09573	0.04922	0.05998

Table F-4 Observational MOM Bias

	w1	w2	m1	m2	s1	s2
n=2500	0.03720	0.03720	0.28502	0.30766	0.15944	0.18201
n=5000	0.01941	0.01941	0.14810	0.16141	0.08416	0.09804
n=7500	0.01390	0.01390	0.10483	0.11727	0.06001	0.07245
n=10000	0.01108	0.01108	0.08406	0.09320	0.04901	0.05716
n=12500	0.00903	0.00903	0.06561	0.07710	0.03713	0.04901
n=15000	0.01013	0.01013	0.07673	0.08455	0.04385	0.05106
n=20000	0.00764	0.00764	0.05628	0.06483	0.03243	0.04063
n=25000	0.00625	0.00625	0.04702	0.05229	0.02721	0.03173
n=30000	0.00586	0.00586	0.04288	0.05000	0.02364	0.03155
n=35000	0.00550	0.00550	0.04000	0.04754	0.02270	0.03117
n=40000	0.00557	0.00557	0.03877	0.04842	0.02232	0.03245
n=45000	0.00404	0.00404	0.02967	0.03441	0.01916	0.02206
n=50000	0.00481	0.00481	0.03506	0.04105	0.01937	0.02580
Average	0.01080	0.01080	0.08108	0.09075	0.04619	0.05578

Table F-5 Actual MPDP Bias

	w1	w2	m1	m2	s1	s2
n=2500	0.03486	0.03486	0.17742	0.13407	0.16253	0.54911
n=5000	0.03278	0.03278	0.15639	0.10162	0.14317	0.54051
n=7500	0.03217	0.03217	0.14741	0.08388	0.14204	0.54443
n=10000	0.03199	0.03199	0.15139	0.07580	0.14456	0.54337
n=12500	0.03276	0.03276	0.14643	0.07334	0.13694	0.54479
n=15000	0.03243	0.03243	0.14347	0.06626	0.13925	0.55138
n=20000	0.03257	0.03257	0.14658	0.06057	0.13970	0.54651
n=25000	0.03127	0.03127	0.13566	0.06607	0.14529	0.55002
n=30000	0.03269	0.03269	0.13691	0.06264	0.15073	0.56093
n=35000	0.03110	0.03110	0.13937	0.05117	0.14592	0.54977
n=40000	0.03341	0.03341	0.15032	0.06339	0.13608	0.56083
n=45000	0.03232	0.03232	0.14274	0.05164	0.14830	0.55584
n=50000	0.03206	0.03206	0.14098	0.05707	0.14582	0.55367
Average	0.03249	0.03249	0.14731	0.07289	0.14464	0.55009

Table F-6 Observational MPDP Bias

	w1	w2	m1	m2	s1	s2
n=2500	0.03391	0.03391	0.17387	0.11999	0.15839	0.55053
n=5000	0.03279	0.03279	0.15481	0.09235	0.14104	0.54192
n=7500	0.03206	0.03206	0.14891	0.07491	0.14060	0.54285
n=10000	0.03194	0.03194	0.14723	0.07089	0.14359	0.54299
n=12500	0.03262	0.03262	0.14706	0.06867	0.13781	0.54671
n=15000	0.03246	0.03246	0.14632	0.06144	0.13816	0.54933
n=20000	0.03241	0.03241	0.14693	0.05974	0.13962	0.54727
n=25000	0.03156	0.03156	0.13386	0.05780	0.14597	0.54823
n=30000	0.03238	0.03238	0.14114	0.06495	0.14838	0.55696
n=35000	0.03151	0.03151	0.13989	0.05060	0.14839	0.54965
n=40000	0.03364	0.03364	0.15152	0.06193	0.13609	0.56211
n=45000	0.03211	0.03211	0.14135	0.05432	0.14856	0.55424
n=50000	0.03219	0.03219	0.13711	0.05322	0.14557	0.55496
Average	0.03243	0.03243	0.14692	0.06852	0.14401	0.54983

F.3. Estimator Rate of Convergence

As discussed in Section 2.3.1, the standard deviation of an estimator decreases exponentially with the sample size. Equation F.3 describes this relationship in several different forms.

$$\begin{aligned}
 \sigma(\hat{\theta}) &\propto m^{-b} \\
 \sigma(\hat{\theta}) &= km^{-b} \\
 \log \sigma(\hat{\theta}) &= -b \log(m) + \log(k) \\
 &= -b \log(m) + K
 \end{aligned}
 \tag{F.3}$$

Figures F-7 through F-9 show the logarithmic scaled plots of the variance of the estimates versus the dataset size for each estimation technique.

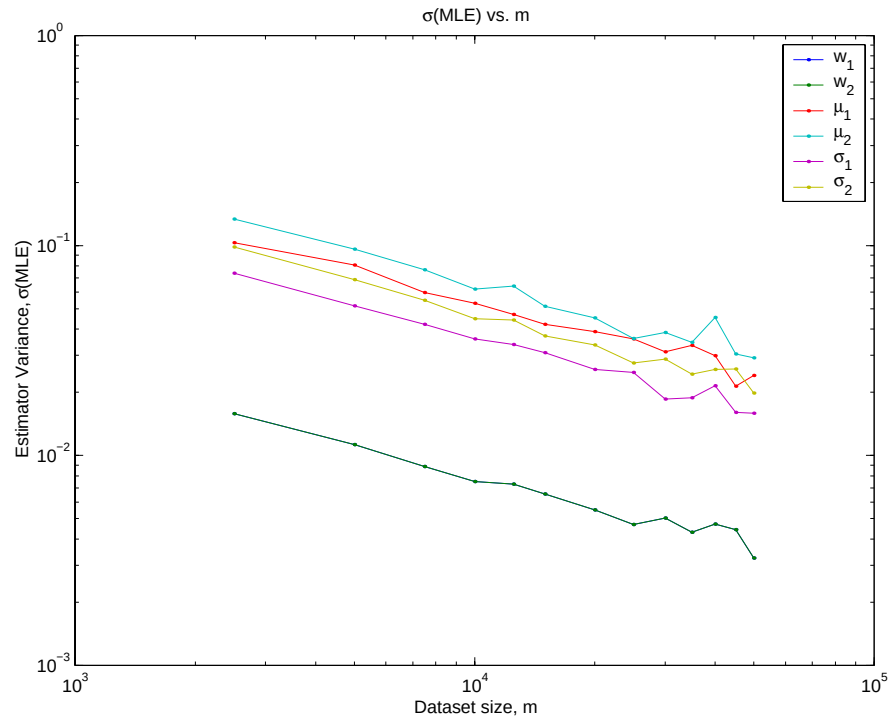


Figure F-7 MLE variance versus data size

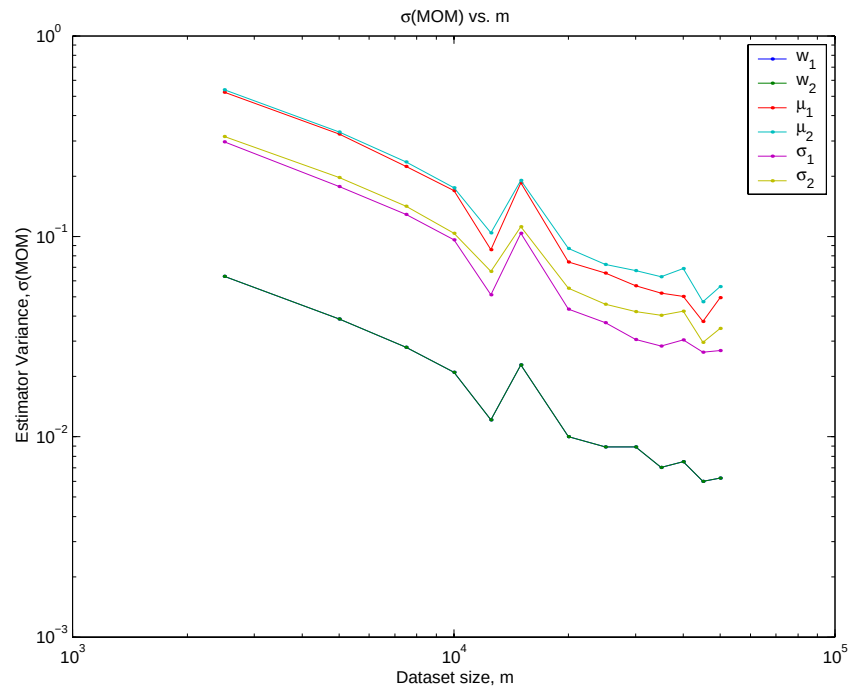


Figure F-8 MOM variance versus data size

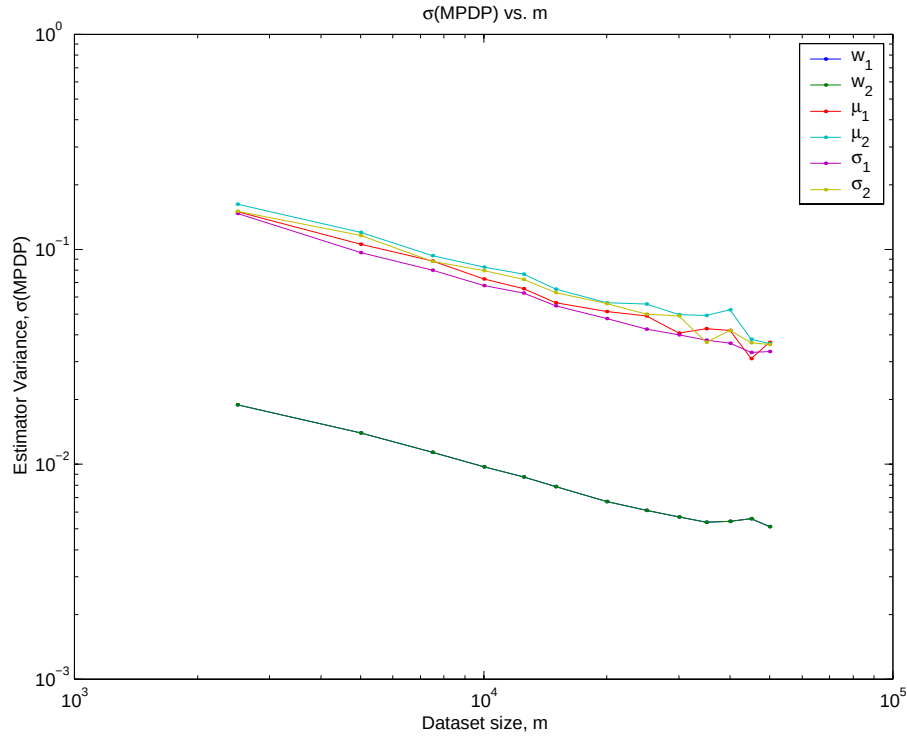


Figure F-9 MPDP Variance versus data size

The final expression of Equation F.3 gives a method of estimating the rate of convergence, performing a least-squares regression on the log of the two variables. The validity of this approach is confirmed by the relative linear trend of the variances on a log-log plot, in Figures F-7 and F-9. Determining the rate of convergence for the method of moments is more problematic. Figure F-8 shows three different trends. For $m < 10,000$, the estimates converge rapidly. For $m > 20,000$, the estimates converge more slowly. The region in between is erratic, perhaps an artifice of the simulation. Since rate of convergence and efficiency are usually referred to in asymptotic terms, i.e, as $m \rightarrow \infty$, the estimates of b are based on the region $m > 20,000$ for the method of moments. Table F-7 summarizes these estimates of the rate of convergence, b .

Table F-7 Estimated Rates of Convergence

	w1	w2	m1	m2	s1	s2
MLE	0.4706	0.4706	0.4884	0.4879	0.5045	0.4977
MOM	0.5618	0.5618	0.5961	0.5115	0.5162	0.5366
MPDP	0.4484	0.4484	0.4890	0.4695	0.4942	0.4926

F.4. Estimator Relative Efficiency

As discussed in Section 2.3.1 and confirmed by Figures F-7 through F-9, the maximum likelihood estimate is expected to be the most efficient, i.e., have the smallest variance. Therefore, the method of moments and the midpoint distance profile technique were compared to it to obtain relative efficiencies. Figures F-10 and F-11 show the relative efficiencies.

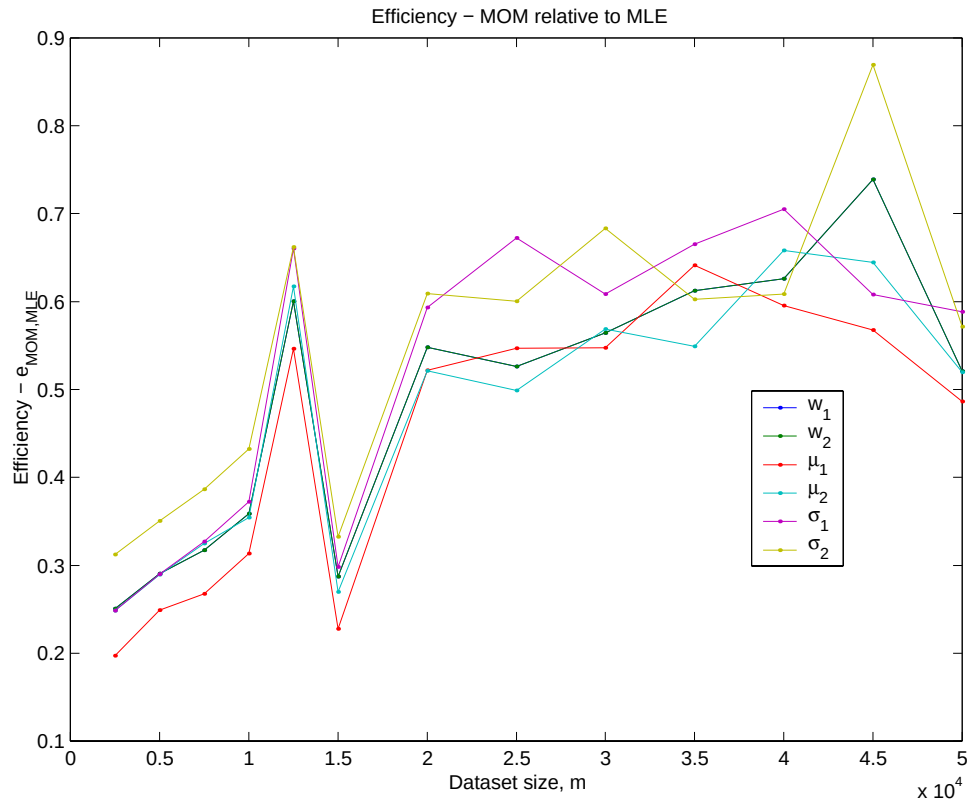


Figure F-10 Relative efficiency of MOM estimation

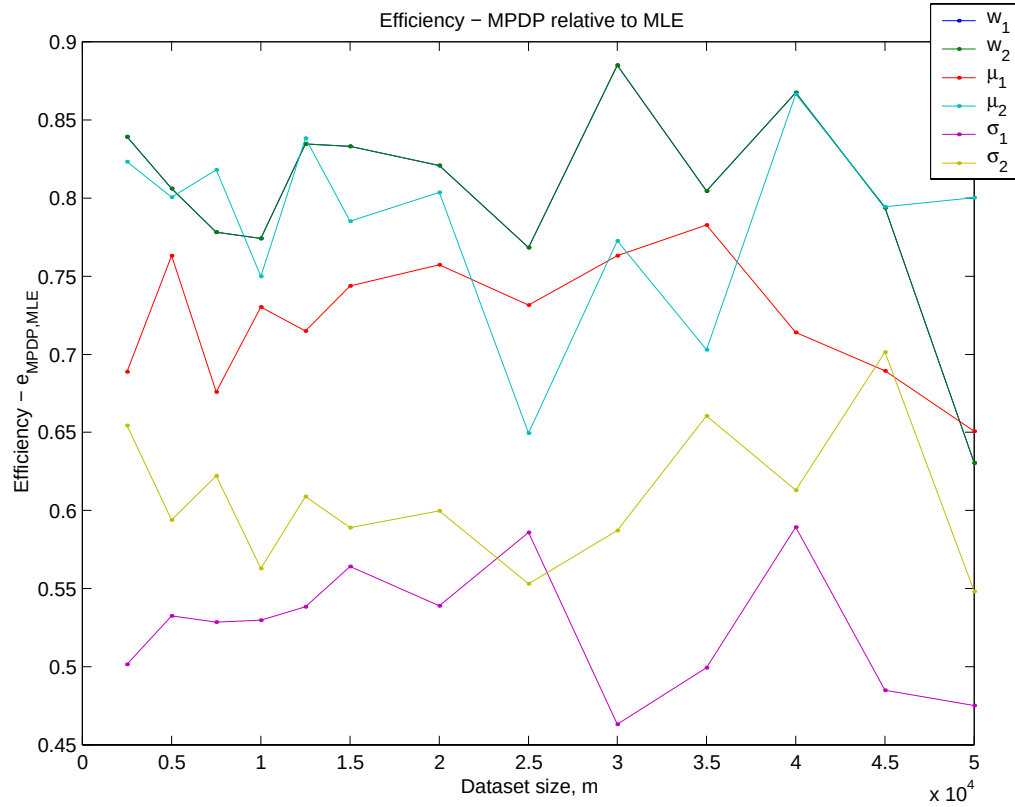


Figure F-11 Relative efficiency of MPDP estimation

Figure F-10 shows a similar trend as Figure F-8. The method of moments estimates appear to have a region of rapidly increasing relative efficiency, followed by an erratic region, followed by a region of relatively stable efficiencies. The asymptotic relative efficiency of the two techniques was loosely approximated as the average of the relative efficiencies over all values of m for the midpoint distance technique and in the stable region for the method of moments. Table F-8 shows these results.

Table F-8 Asymptotic Relative Efficiencies

	w1	w2	m1	m2	s1	s2
$E\{\text{mom}, \text{mle}\}$	0.5910	0.5910	0.5582	0.5658	0.6345	0.6493
$E\{\text{mpdp}, \text{mle}\}$	0.8028	0.8028	0.7235	0.7850	0.5256	0.6073

Vita

Carl Edward Minton was born April 29, 1975 in Waynesboro, Virginia. In 1993, he graduated as valedictorian from Riverheads High School in Staunton, Virginia. After high school, he matriculated at Virginia Polytechnic Institute and State University in Blacksburg, Virginia. In 1997 he graduated summa cum laude with a degree in Electrical Engineering and a minor in Political Science. From 1997 until 1999 he pursued his graduate education at Virginia Tech, receiving the Bradley Fellowship from the Department of Electrical and Computer Engineering. After a brief career misdirection (law school, who was he kidding?) in 1999 he accepted an analyst position with the Information Technology and Communication department of the University of Virginia. On August 11, 2001 he was married to Amy Karen Rider, of Watsonville, California. He and his wife currently reside in Charlottesville, Virginia, where he remains in the employ of the University of Virginia.