CONTINUOUS MIXTURE MODELING
VIA
GOODNESS-OF-FIT CORES

by

Stephen Ronald Aylward

A dissertation submitted to the faculty of The University of North Carolina at Chapel Hill
in partial fulfillment of the requirements for the degree of Doctor of Philosophy
in the Department of Computer Science.

Chapel Hill
1997

Approved by

Advisor: Dr. James Coggins

Reader: Dr. Stephen Pizer

Dr. Daniel Fritsch

Dr. Steve Marron

Dr. Jonathan Marshall

Dr. Keith Muller

ABSTRACT

Stephen Ronald Aylward:
CONTINUOUS MIXTURE MODELING VIA GOODNESS-OF-FIT CORES
(Under the direction of Dr. James Coggins)

This dissertation introduces a new technique for the automated specification of continuous Gaussian mixture models (CGMMs). This approach is ideal for representing distributions that arise in a variety of problems including the driving problem of this dissertation, representing the distribution of the intensities associated with tissues in medical images.

Gaussian mixture models are population density representations formed by the weighted linear combination of multiple, multivariate Gaussian component distributions. Finite Gaussian mixture models are formed when a finite number of discrete component distributions are used. CGMMs are formed when multiple continua of component distributions are used.

The approach to CGMM specification developed here is based on an original structure, the Gaussian goodness-of-fit (GGoF) core. GGoF functions quantify how well a Gaussian having a particular mean and covariance represents the local distribution of samples. GGoF cores capture the continua of Gaussians which well represent a sampled population; they define a CGMM. In this dissertation, Monte Carlo simulations are used to evaluate the robustness of a variety of GGoF functions and binning methods for the specification of GGoF cores. The log likelihood ratio GGoF function is shown to produce the most accurate and consistent GGoF cores.

In generalized projective Gaussian distributions, similar feature vectors, when projected onto a subset of basis feature vectors, have a Gaussian distribution. In this dissertation, Monte Carlo simulations and ROC analysis are used to demonstrate that for such distributions CGMMs defined using GGoF cores produce accurate and consistent labelings compared to K-means and finite Gaussian mixture models. Additionally, CGMMs do not suffer from the problems associated with determining an appropriate number of components, initializing the component parameters, or iteratively converging to a solution.

Generalized projective Gaussian distributions exist in a variety of real-world data. The applicability of CGMM via GGoF cores to real-world data is demonstrated through the accurate modeling of tissues in an inhomogeneous magnetic resonance image. The extension of CGMM via GGoF cores to high-dimensional data is demonstrated through the accurate modeling of a sampled trivariate anisotropic generalized projective Gaussian distribution.

DEDICATION

To my family for the opportunity and encouragement to pursue my dreams.

Thanks also goes to numerous people at McDonnell Douglas
who provided me with the initiative to start this work
and to the many coffee shops of Chapel Hill
in which most of this work was completed.

TABLE OF CONTENTS

LIST OF SYMBOLS

General notation convention:

F()	Function	Capital bold followed by parenthesis
S	Set	Capital italics
$\underline{x}$	Vector	Single underscore
$\underline{x}_i$	Vector component i	Subscripted vector
$\underline{\underline{\bullet}}$	Matrix	Double underscore
$\underline{\underline{\bullet}}_{ij}$	Matrix component ij	Double subscripted matrix
$\underline{x}^{(i)}$	ith instance	A vector, set, etc. with superscript in parenthesis
$\vert S \vert$	Size of Set S	
$\vert \underline{\underline{\bullet}} \vert$	Determinant of Matrix	
$\vert \vert \underline{v} \vert \vert$	Length of Vector	
$\underline{v}\,'$	Transpose of Vector	

Symbols:

B	Number of bins
f_0	Feature 0
f_1	Feature 1
$\underline{\underline{H}}$	Hessian
K	Number of components in a finite mixture model
M	The dimension of a ridge
$\underline{\mu}$	Mean vector
N	Number of features comprising a sample (i.e., the dim. of feature space)
N_c	Number of classes/populations in a pattern recognition problem
N_t	Number of tracks comprising the CGMM
$\underline{\underline{P}}$	Projection matrix used by multivariate binning processes
Q	Numer of directions of projection: Rank of $\underline{\underline{P}}$
R	Number of Monte Carlo runs in an experiment
ρ	Constant of proportionality between scale and radius

r	radius
s	Size parameter for multidimensional GGOF functions
σ	standard deviation, scale
$S^{(\mathrm{tr})}$	Set of all training samples
$S^{(\mathrm{tr:A})}$	Set of all training samples from Class A
$\underline{x}$	A sample or spatial point
$\underline{\underline{\bullet}}$	Covariance matrix
$\underline{\underline{\bullet}}(\underline{\mu},s)$	Local data’s covariance matrix
X^2	A Chi-squared based GoF function
X_{P}^2	Pearson’s Chi-squared GoF function
$X_{\mathrm{R\&C}}^2$	Read & Cressie’s power divergent GoF function
X_{LLR}^2	Log likelihood ratio GoF function
$\chi_{\mathrm{B}-1}^2(\alpha)$	Value from a Chi-squared table, B-1 degree of freedom, α power

LIST OF ABBREVIATIONS

CGMM	Continuous Gaussian Mixture Modeling
FGMM	Finite Gaussian Mixture Modeling
FPR	False-positive rate
GOF	Goodness-of-fit
GGOF	Gaussian goodness-of-fit
GMM	Gaussian mixture modeling
KM	K-Means
KM7	K-Means using 7 components per population
KNN	K-Nearest Neighbor
KNN3	K-Nearest Neighbor using 3 nearest neighbors
LoG	Laplacian of Gaussian
MLEM	Maximum likelihood expectation maximization
MLP	Multilayered perceptron
MLP6x3	Multilayered perceptron having two hidden layers with 6 nodes in the first hidden layer and 3 in the second hidden layer.
PW	Parzen window
PW2	Parzen window with Gaussian kernel with a standard deviation of 2 units
TPR	True-positive rate

Chapter 1

INTRODUCTION

Normality is a myth
there never was, and never will, be a normal distribution.
- Geary, 1947

If the clusters are compact and isolated,
almost any representation will work.
- Coggins, 1996

1.1. Statistical Pattern Recognition

Most scientists encounter problems which involve statistical analysis. What populations are present in my data? How do these populations differ? Have I collected enough data? From which population did this sample originate? In pattern recognition systems, samples are used to form models of their source populations. This dissertation introduces a novel technique for consistently and accurately forming those models. Questions such as those above are answered using measurements derived from such models. [Duda and Hart 1973; Schalkoff 1992]

Two types of pattern recognition systems are clusterers and classifiers. They are distinguished by whether the training samples used for determining their parameters are required to have labels.

Clusterers do not use labels during the formation of their models. Clusterers attempt to partition a set of training samples into groups (i.e., clusters) of similar samples. Each cluster is then assigned a unique label, or the labeled samples in each cluster vote to determine an appropriate cluster label.

Classifiers require that each sample in the training set have a label indicating its source population. These labels limit a sample's influence to the parameters of its own population's model. This dissertation is mainly concerned with the development of population models for use in classifiers.

A sample is an instance of N measurements obtained from an object. A sample captures the "features" of an object and maps that object to a point in an N-dimensional "feature space."

A population is a source or category or class from which objects originate. Presumably, every object associated with a population will share a set of traits that are characterized by the measurements used to define the samples.

For statistical pattern recognition, samples are manipulated as vectors of N random variables. Variations among samples from the same population are indicative of noise and/or a lack of correspondence between the measurements and the common traits. This variance and the correlations in the measurements determine how a population's samples will be distributed in feature space.

Pattern recognition systems attempt to model, via an implicitly or explicitly estimated density function, the distribution of a population's samples in feature space. While feature quality limits the potential accuracy of a pattern recognition system, it is the estimated density function which determines the accuracy and consistency actually achieved. The focus of this dissertation is the accurate and consistent specification of density functions common to a variety of distributions including those occurring in medical imaging, speech recognition, and handwriting recognition.

1.2. Density Functions

Given a sample in feature space, a density function for a class provides the probability that that sample originated from that class. Density functions are distinguished by the assumptions they impose and their parameter specification ("training") process.

A frequently used density function assumes that a population's distribution is multivariate normal, i.e., Gaussian. Gaussian density functions are completely parameterized by a mean vector and a covariance matrix. While Geary [Geary 1947] and others may claim that Gaussian assumptions are rarely correct, they have been shown to be applicable to many real-world populations, e.g., magnetic resonance imaged tissue intensity distributions after removal of spatially correlated intensity variations [Dawant, Zijdenbos et al. 1993; Aylward and Coggins 1994; Meyer, Bland et al. 1995; Wells III, Grimson et al. 1996]. In general, when a Gaussian density function is assumed and that assumption is correct, the corresponding pattern recognition system produces consistent and optimally accurate labelings with respect to the features being used. Additionally, as stated by Coggins[Coggins 1996], even when the assumed density function is not correct, the separation of the populations in feature space may be large enough that a suboptimal representation can provide sufficient accuracy for the problem at hand. For many long-standing pattern recognition problems, however, the Gaussian assumption has been shown to be incorrect and insufficient, and a sufficiently accurate density function assumption is not known. In some of these situations, improved pattern recognition accuracy has resulted from the development of Gaussian mixture model (GMM) density functions [Bellegarda and Nahamoo 1990; Aylward and Coggins 1994; Bellegarda, Bellegarda et al. 1994; Gish and Schmidt 1994; Zhuang, Huang et al. 1996].

1.3. Gaussian Mixture Models

A mixture model is formed by a weighted linear combination of multiple "component" distributions. Its parameters, Ψ, include the *a priori* probabilities of the components $\omega^{(i)}$ and the parameters of the individual components $\Phi^{(i)}$. In a GMM, the component distributions, $\mathbf{F}(\underline{x};\phi^{(i)})$, are multivariate normal densities; each component is an N-dimensional Gaussian distribution parameterized by a mean $\underline{\mu}$ and covariance matrix $\underline{\underline{\Sigma}}$.

$$\mathbf{F}\left(\underline{x};\Phi^{(i)}\right)=\frac{1}{(2\pi)^{N/2}\left|\underline{\underline{\Sigma}}\right|^{1/2}}e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}\right)^{t}\underline{\underline{\Sigma}}^{-1}\left(\underline{x}-\underline{\mu}\right)} \qquad \Phi^{(i)}=\left\{\underline{\mu},\underline{\underline{\Sigma}}\right\} \qquad [1.1]$$

where ^{t} denotes transposition
$|\underline{\underline{\bullet}}|$ denotes the determinant of the matrix $\underline{\underline{\bullet}}$
single underscore denotes a vector
double underscore denotes a matrix

If the number of components, K, is bounded, the mixture model is called a finite mixture model. A finite mixture model provides a probability for a sample $\underline{x} \in \Re^{N}$ via

$$\mathbf{P}\left(\underline{x}\mid\Psi\right)=\sum_{i=1}^{K}\omega^{(i)}\,\mathbf{F}\left(\underline{x},\Phi^{(i)}\right) \qquad [1.2]$$

where

$$1=\sum_{i=1}^{K}\omega^{(i)} \qquad \text{and} \qquad \Psi=\left\{\{\omega,\Phi\}^{(i)}\mid i=1..K\right\} \qquad [1.3]$$

If the components are defined as spanning one or more tracks through their parameter space, i.e., the domain of the parameter's of a component, then the model is formed by the combination of an infinite number of continuously varying components and thus is called a continuous mixture model. That is, a continuous mixture model consists of components whose parameters are spanned by one of N_t continua of points, $\mathbf{T}^{(j)}$, through the model's parameter space. A continuous mixture model provides a probability for a sample $\underline{x} \in \Re^N$ via

$$\mathbf{P}\left(\underline{x} \mid \mathit{\Psi}\right) = \underset{\{\omega, \mathit{\Phi}\} \in \mathit{\Psi}}{\mathbf{MAX}}\left(\omega\,\mathbf{F}\left(\underline{x}, \mathit{\Phi}\right)\right) \qquad [1.4]$$

where

$$\mathit{\Psi} = \left\{\{\omega, \mathit{\Phi}\} \,\middle|\, \exists\, j \in 1..N_t \text{ s.t. } \mathit{\Phi} \in \mathbf{T}^{(j)} \text{ and } \omega = \mathbf{P}(\mathit{\Phi})\right] \qquad [1.5]$$

Equation 1.4 states that each sample is in fact generated by just one of the infinite number of components and the generating component is determined via maximum likelihood and that component provides the best estimate of the sample's probability. Thus $\mathbf{F}(\underline{x},\phi)$ can be interpreted as a providing a point conditional sample probability, and ω as providing a point *a priori* probability. Equation 1.4 can therefore be rewritten as

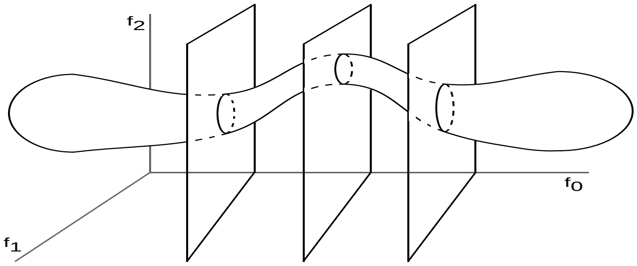
$$\mathbf{P}\left(\underline{x} \mid \mathit{\Psi}\right) = \underset{\{\mathit{\Phi}\} \in \mathbf{T}^{(j)} | j=1..N_t}{\mathbf{MAX}}\left(\mathbf{P}(\mathit{\Phi})\,\mathbf{P}\left(\underline{x} \middle| \mathit{\Phi}\right)\right) \qquad [1.6]$$

The focus of this dissertation is the definition of the continua of points ϕ $\mathbf{T}^{(j)}$ via core techniques and the estimation of their associated $P(\phi)$ using traditional statistical methods.

1.4. Why Gaussian Mixture Models

This dissertation introduces the concept of a generalized projective Gaussian distribution. If the projection of a group of similar samples onto a subset of basis directions has a Gaussian distribution, those samples are said to have a generalized projective Gaussian distribution (Figure 1.1). The term "extruded" is generalized to refer to the stretching and the scaling these distributions can exhibit. Such "extruded" Gaussian distributions occur when correlations exist between a population's parameters and some of the population's features.

Independently, [Gerig, Martin et al. 1991; Aylward and Coggins 1994; Wells III, Grimson et al. 1996] demonstrated that the intensities associated with individual tissue types in an MR image have non-Gaussian distributions. Yet, tissue samples within small regions in an MR image



An isoprobability surface of an extruded Gaussian distribution
Figure 1.1

have been shown to be well represented using a Gaussian distribution, and the parameters of these spatially localized Gaussian distributions have been shown to vary smoothly across the image [Aylward and Coggins 1994; Wells III, Grimson et al. 1996]. Thus a continua of Gaussians well models those spatial variations and represents each distribution.

In speech recognition, it is commonly accepted that hidden Markov models based on Gaussian distributions can represent the speech of a single person in a controlled situation, e.g., given a fixed level of stress, background noise, etc. Additionally, smooth warps can be applied to the parameters of those Gaussians to adapt them to new situations and speakers [Bellegarda and Nahamoo 1990]. Thus, to account for variations in speaker and situation, multiple Gaussians are needed.

For some applications the feature/model-parameter correlations are well understood and easily measured. In those situations, the most accurate labelings can be obtained by directly eliminating their effects and then using a simple Gaussian classifier [Axel, Costantini et al. 1987; Brey and Narayana 1988; Dawant, Zijdenbos et al. 1993; Aylward and Coggins 1994; Meyer, Bland et al. 1995; Wells III, Grimson et al. 1996]. However, when the correlations are not well understood or easily measured, GMMs are appropriate.

1.5. Why not Finite Gaussian Mixture Models?

Most investigations involving GMMs have used FGMMs. The development of an algorithm for fast, accurate, and consistent FGMM training has been the focus of most GMM research. A concise history of this research can be found in Section 2.3.7. A more detailed history is given in [Titterington, Smith et al. 1985] and [McLachlan and Basford 1988].

While clearly no single training algorithm is best in all situations, maximum likelihood expectation maximization (MLEM) provides several desirable convergence properties, e.g., monotonic convergence rate [Titterington, Smith et al. 1985], and is easy to implement and use. MLEM, however, is an approximate gradient ascent algorithm, and maximum likelihood is subject to local maxima and non-optimal global maxima [Zhuang, Huang et al. 1996]. When gradient ascent is applied to a function having local maxima, the results are dependent on the algorithm's initial conditions. While MLEM is less likely to settle into these local maxima compared to other FGMM training algorithms [Jordan and Xu 1993], it will be shown in Section 2.3.7.3 that for a given pattern recognition problem, FGMM component configurations generated via MLEM can vary greatly and be far from optimal due to the local maxima. This difficulty is aggravated by the reliance on the user to specify the number of components. While much research has focused on automatically determining the appropriate number of components [McLachlan and Basford 1988; West 1993; Zhuang, Huang et al. 1996], McLachlan states that "testing for the number of components...in a mixture is an important but very difficult problem which has not been completely resolved." For generalized projective Gaussian distributions, there are actually an infinite number of components, so deciding an appropriate finite number of components to approximate such distributions can be especially difficult. Thus, although GMMs are well suited for a variety of pattern recognition problems, FGMMs via MLEM can provide poor labeling consistency due to their reliance on the user to specify an appropriate number of components, the initialization of its parameters, and the particular collection of samples used in training.

1.6. Why Continuous Gaussian Mixture Models (Thesis)

Extruding a Gaussian distribution can be visualized as producing a track of means central to the distribution along which the variance of the data normal to that track changes smoothly. This dissertation uses a novel mechanism, GGoF cores, to approximate that central track of means and determine the local variance of the data. Those tracks of means and their variances define a CGMM. The author of this dissertation asserts that

> *In Monte Carlo studies against competing techniques, i.e., K-means and finite Gaussian mixture modeling, the proposed method is more automated, more accurate, and as consistent when representing generalized projective Gaussian distributions.*

That is, a CGMM of an extruded Gaussian distribution is accurately and consistently defined using a Gaussian-goodness-of-fit (GGoF) function and a process for tracking the generalized maxima of this function, i.e., core extraction. The user is not required to specify a hyperparameter such as the number of components, and if multiple cores of a distribution are extracted, they will serve to refine and not confound the representation.

The accuracy and consistency of these distribution representations are quantified by the accuracy and consistency of the classifiers they are used to define. Classifier accuracy is measured via true positive and false positive classification rates. The consistency of a classifier is quantified by the variance of its accuracy over a series of Monte Carlo simulations.

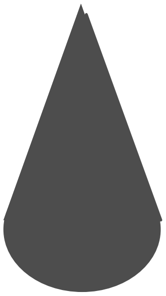
1.7. Object Representation via Medialness Cores

This dissertation extends the notion of cores from medial traces in scale space [Morse, Pizer et al. 1996; Pizer, Eberly et al. 1996] to the representation of a sampled distribution. To explain this domain shift, this section provides a brief introduction to medialness cores by defining medialness functions and medialness space.

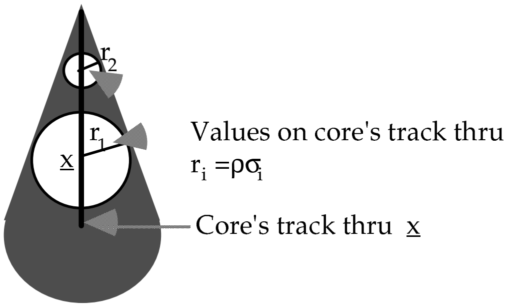
Medialness functions: A real medialness function $\mathbf{M}(\underline{x},\sigma)$ responds particularly strongly when it is applied at a point, $\underline{x}$, on the central skeleton of an object using an accurate local object scale estimate, σ. The scale, σ, is proportional to the radius, r, of the maximally inscribed circle centered at $\underline{x}$ (i.e., $r = \rho\sigma$). (Section 3.1)

Medialness Space: A medialness space is formed by computing a medialness function on an image over a range of $\underline{x}$ and σ values. Thus, an N-dimensional image yields an (N+1)-dimensional medialness space. (Section 3.2)

Medialness Cores: Medialness cores capture the location, size, and shape of objects in an image. They exist in the medialness space as traces of *generalized maxima* in medialness. That is, they define traces in medialness space such that the points on a trace are local maxima as measured in directions normal to the trace. Medialness cores have been applied to a wide range of objects in a variety of images. Figures 1.2 and 1.3 depict a binary object and its core. (Section 3.3)



The binary image of an object
Figure 1.2


r_2
r_1
$\underline{x}$
Values on core's track thru
$r_i = \rho\sigma_i$
Core's track thru $\underline{x}$

The spatial projection of its medialness core and two r values on that core's track
Figure 1.3

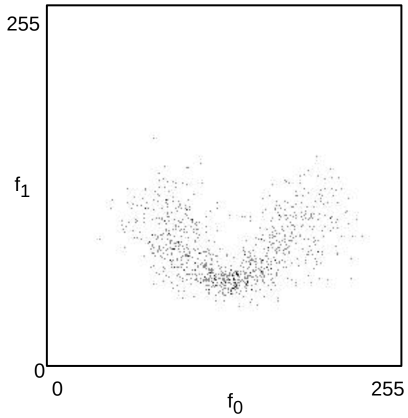
1.8. Gaussian Mixture Modeling via Gaussian-Goodness-of-Fit Cores

This dissertation is based on the realization that just as cores of medialness functions can represent the shapes of objects in images, cores of GGoF functions can represent the shapes of distributions in feature spaces. Instead of selecting points in medialness space to fit an object's shape, GGoF cores select parameterizations of Gaussians to represent a population's density function. Thus, GGoF cores define CGMMs. GGoF cores exist in a GGoF space and capture the location, size, shape, and density of distributions in feature space. CGMMs via GGoF cores are ideal for representing generalized projective Gaussian distributions. The basic concepts of the contributing technologies are presented below.

Feature Space: Feature space is the domain of samples. Each sample, comprised of N random variables, exists at a single point in an N-dimensional feature space. Feature space is also the domain of density function estimates and decision bounds which are used in every pattern recognition system. Thus, feature space is a pattern recognition system's view of the problem at hand. (Section 2.1.2.)

Scattergrams: Scattergrams allow us to view the distribution of samples in feature space. Just as a histogram depicts the frequency of a value (or a range of values) of a single random variable using height in a bar graph, a scattergram is an image that depicts the frequency of an N-variate sample's values as intensity (best if N•3). The scattergram of a collection of "similar" samples will contain a cluster or cloud of high intensity. Figure 1.4 depicts a scattergram of a collection of 900 samples from a simulated generalized projective Gaussian population (to be detailed in Chapter 2). Scattergrams can also be used to visualize the action of a pattern recognition system. Density functions can be depicted via intensity distributions or isoprobability curves, decision bounds can be shown as hypersurfaces, and decision regions can be shown using coloring. (Sections 2.1.3 and 2.1.4.)

Gaussian-Goodness-Of-Fit Functions: A Gaussian-goodness-of-fit (GGoF) function quantifies how well a Gaussian with a particular mean, $\underline{\mu}$, and covariance, $\underline{\underline{\Sigma}}$, represents the distribution of samples within a local region of feature space, e.g., within $\pm 2\underline{\underline{\Sigma}}$ of $\underline{\mu}$ [Cressie and Read 1984; Read and Cressie 1988; Rayner and Best 1989]. That is, they quantify how well a



The scattergram of a collection of samples
Figure 1.4

Gaussian represents a population's local density function. For a population having a Gaussian distribution, GGoF functions respond maximally when they are evaluated at the population's actual mean and covariance. Commonly used instances of this class of functions include Pearson's Chi-squared, the log likelihood ratio, and Kolmogorov-Smirnov functions [Stephens 1974; D'Agostino and Stephens 1986]. (Section 4.1.)

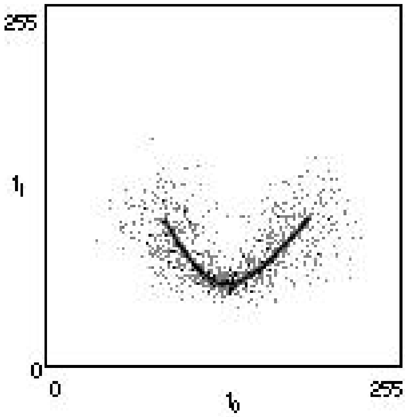
GGoF space is formed from the application of a GGoF function to a collection of samples for a range of Gaussian parameter values. By defining the Gaussian's mean to be of dimension N and the Gaussian's covariance matrix $\underline{\underline{\Sigma}}$ to be a function of a single parameter s, an N-dimensional scattergram has a corresponding (N+1)-dimensional GGoF space. (Section 5.1.)

GGoF Cores: The traces of generalized maxima of a GGoF function are GGoF cores. They define traces in GGoF space such that the points on each trace are local maxima in the directions normal to that trace. These traces $\mathbf{T}^{(j)}$ in $\{\underline{\mu}, \underline{\underline{\Sigma}}(s)\}=\phi$ along with a local sample frequency estimate ω specify the $P(\underline{x} \mid \phi)$ and $P(\phi)$ which define a CGMM. (Section 5.3.)

Generalized Projective Gaussian Distributions: When the distribution of the samples is generalized projective Gaussian the GGoF core produces an accurate representation of the distribution. That is, when a distribution in an N-dimensional feature space is projected onto an (N–M)-dimensional hyperplane spanning the subset of basis projective Gaussian directions, an (N–M)-dimensional Gaussian distribution results. When those directions correspond to the directions for which the GGoF function is locally maximal, i.e., the core normal directions, the placement of a Gaussian component at the $\underline{\mu}$ and $\underline{\underline{\Sigma}}(s)$ of a GGoF core point will form a representation which is a locally optimal and accurate fit to the distribution.

When the distribution of the samples is not generalized projective Gaussian in the (N–M) directions normal to the core, the representation may be sufficiently accurate for the problem at hand, "Coggins' rule."

The sampled population in Figure 1.4 has a generalized projective Gaussian distribution.



Projection of the feature space component of a GGoF core of a generalized projective Gaussian distribution onto its scattergram
Figure 1.5

Figure 1.5 depicts the feature space component of a GGoF core of that distribution. This dissertation is concerned with the extraction of these cores, the use of these cores in the definition of CGMM representations of a population's distribution, and assessing the accuracy and consistency of those CGMM representations.

1.9. Synopsis

This dissertation demonstrates, using both simulated and "real-world" data, that a CGMM of a generalized projective Gaussian distribution can be defined using GGoF cores. When such models are used for classification, accurate labelings are produced. Experiments described herein indicate that for small false positive rates, CGMMs via GGoF cores provide superior true positive rates compared to K-means and FGMM. These labelings are consistent, and the extraction of multiple GGoF cores improves the accuracy and consistency of the models. CGMMs avoid reliance on the user for the specification of a hyperparameter such as K for the number of components and the problems associated with local maxima in the iterative parameter refinement process.

Chapter 2 provides an overview of several popular classification methods. Special attention is given to explaining finite Gaussian mixture modeling and assessing its accuracy and consistency when used to define a classification system. Chapter 2 is also used to detail and motivate two related two-feature, two-class pattern recognition problems which are used in comparisons and explanations throughout this dissertation.

Chapter 3 provides an overview of medialness core definition, extraction, and application.

Chapter 4 introduces goodness-of-fit functions and evaluates the accuracy and consistency with which they can identify the parameters of a univariate Gaussian and a univariate skewed Gaussian distribution.

Chapter 5 details the process of continuous Gaussian mixture model definition and operation using Gaussian goodness-of-fit cores.

Chapter 6 reports the results of a controlled study which compares continuous Gaussian mixture modeling via GGoF cores with K-means and finite Gaussian mixture modeling. Comparisons are also made between these methods using an inhomogeneous magnetic resonance image and a trivariate extruded elliptical Gaussian distribution.

Chapter 7 contains summaries of the major contributions and areas for future research.

1.10 Bibliography

Axel, L., J. Costantini, et al. (1987). "Intensity Correction in Surface-Coil MR Imaging." American Journal of Radiology **148**: 418-420.

Aylward, S. R. and J. M. Coggins (1994). Spatially Invariant Classification of Tissues in MR Images. Visualization in Biomedical Computing, Rochester, MN

Bellegarda, E., J. Bellegarda, et al. (1994). "A Fast Statistical Mixture Algorithm for On-Line Handwriting Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(12): 1227.

Bellegarda, J. R. and D. Nahamoo (1990). "Tied Mixture Continuous Parameter Modeling for Speech Recognition." IEEE Transactions on Acoustics, Speech, and Signal Processing **38**(12): 2033-2045.

Brey, W. W. and P. A. Narayana (1988). "Correction for Intensity Falloff in Surface Coil Magnetic Resonance Imaging." Medical Physics **15**(2): 241-245.

Coggins, J. (1996). Conversation.

Cressie, N. and T. R. C. Read (1984). "Multinomial goodness-of-fit tests." Journal of the Royal Statistical Society **46**(4): 440-464.

D'Agostino, R. B. and M. A. Stephens (1986). Goodness-of-Fit Techniques. New York, Marcel Dekker, Inc.

Dawant, B. M., A. P. Zijdenbos, et al. (1993). "Correction of Intensity Variations in MR Images for Computer-Aided Tissue Classification." IEEE Transactions on Medical Imaging **12**(4): 770-781.

Duda, R. and P. Hart (1973). Pattern Classification and Scene Analysis. New York, John Wiley and Sons.

Geary, R. C. (1947). "Testing for normality." Biometrika **34**: 209-242.

Gerig, G., J. Martin, et al. (1991). Automating Segmentation of Dual-Echo MR Head Data. Information Processing in Medical Imaging,

Gish, H. and M. Schmidt (1994). "Text-Independent Speaker Identification." IEEE Signal Processing Magazine **11**(4): 18-32.

Jordan, M. I. and L. Xu (1993). Convergence Results for the EM Approach to Mixtures of Experts Architectures. Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

McLachlan, G. J. and K. E. Basford (1988). Mixture Models. New York, Marcel Dekker, Inc.

Meyer, C. R., P. H. Bland, et al. (1995). "Retrospective Correction of Intensity Inhomogeneities in MRI." IEEE Transactions on Medical Imaging **14**(1): 36-41.

Morse, B. S., S. M. Pizer, et al. (1996). "Zoom-Invariant Vision of Figural Shape: Effects on Cores of Image Disturbances." Computer Vision and Image Understanding *Submitted*

Pizer, S. M., D. Eberly, et al. (1996). "Zoom-invariant Vision of Figural Shape: the Mathematics of Cores." Computer Vision and Image Understanding *Submitted*

Rayner, J. C. W. and D. J. Best (1989). Smooth Tests of Goodness of Fit. Oxford, Oxford University Press.

Read, T. R. C. and N. A. C. Cressie (1988). Goodness-of-fit statistics for discrete multivariate data. New York, Springer-Verlag.

Schalkoff, R. (1992). Pattern recognition: statistical, structural and neural approaches. New York, John Wiley & Sons, Inc.

Stephens, M. A. (1974). "EDF Statistics for goodness of fit and some comparisons." Journal of the American Statistical Association **69**(347): 730-737.

Titterington, D. M., A. G. M. Smith, et al. (1985). Statistical Analysis of Finite Mixture Distributions. Chichester, John Wiley and Sons.

Wells III, W. M., W. E. L. Grimson, et al. (1996). "Adaptive Segmentation of MRI Data." IEEE Transactions on Medical Imaging **15**(4): 429-442.

West, M. (1993). "Approximating Posterior Distributions by Mixtures." Journal of the Royal Statistical Society **55**(2): 409-422.

Zhuang, X., Y. Huang, et al. (1996). "Gaussian Mixture Density Modeling, Decomposition, and Applications." IEEE Transactions on Image Processing **5**(9): 1293-1302.

Chapter 2

CLASSIFICATION

The art of being wise is the art of knowing what to overlook.

- James, 1890

This chapter discusses several popular methods for classification. Finite Gaussian mixture modeling is one such method, and it is presented in detail. Two related two-feature (two-dimensional), two-class pattern recognition problems are used to illustrate the operation of these methods and quantify their performance in terms of the accuracy and consistency with which these methods label new samples.

2.1. Traditional Pattern Recognition

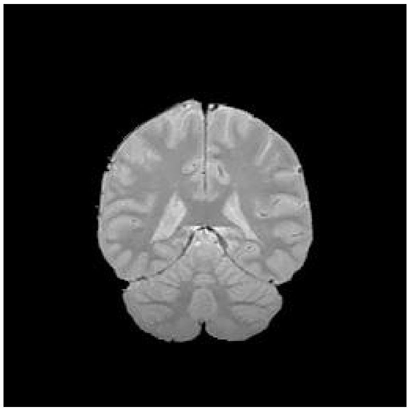
This section explains the concepts necessary for understanding the operating characteristics, strengths, and weaknesses of the density estimation components of several common classification systems. Direct comparisons between finite Gaussian mixture models and other techniques are made to motivate the selection of Gaussian mixture modeling as the focus of this dissertation. Two related, two-class, two-feature classification problems are used for illustration.

2.1.1. Two Related Two-Feature, Two-Class Problems

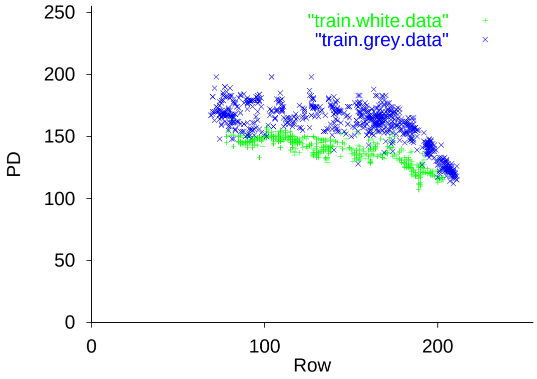
For the visualization and analysis of the operation and performance of the pattern recognition systems discussed in this dissertation, two related two-feature (N=2), two-class (N_c=2) problems are presented. For both problems, the two features are designated f_0 and f_1.

These features are discrete; they can attain any integer value from 0 to 255. The distributions in these problems are motivated by the intensity distributions of tissues in inhomogeneous magnetic resonance (MR) images.

Consider the proton density MR image shown in Figure 2.1. It contains an inhomogeneity which is revealed by a dimming in the inferior cerebellum (lower portion of the brain). A scatterplot of 984 hand-labeled white matter samples and 788 hand-labeled gray matter samples from this image are shown in Figure 2.2. The effect of the inhomogeneity is clear.



Proton Density (PD) MR image
Figure 2.1



Scatterplot of samples from Figure 2.1
Figure 2.2

2.1.1.1. Problem 1 Description and Justification

For Problem 1, the two populations are designated Class A and Class B.

Class B is designed to mimic the intensity distribution of one tissue type in an inhomogeneous MR image. It is thus a generalized projective Gaussian distribution. It is defined by three cubic B-splines [Press, Flannery et al. 1990] and four isotropic control Gaussians, i.e., $G^{(0)} \ldots G^{(3)}$. Each spline governs one of the three parameters of the continua of Gaussians, i.e., one spline for each component of the mean vector and one for the variance. The parameters of the control Gaussians are given in Table 2.1. A visualization of the control Gaussians' density functions with the track of $\underline{\mu}^{(t)}$ for t [0,1] overlaid is shown in Figure 2.3.

Control 0:	**f_0**	**f_1**
$G^{(0)}$		
Mean	80	112
Covar	324	0
f_0		
f_1	0	324

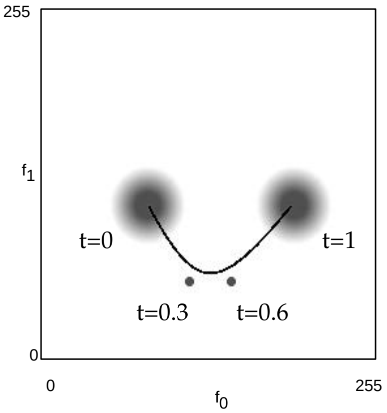
Control 1:	**f_0**	**f_1**
$G^{(1)}$		
Mean	112	56
Covar	1	0
f_0		
f_1	0	1

Control 2:	**f_0**	**f_1**
$G^{(2)}$		
Mean	144	56
Covar	1	0
f_0		
f_1	0	1

Control 3:	**f_0**	**f_1**
$G^{(3)}$		
Mean	192	112
Covar	324	0
f_0		
f_1	0	324

The parameters of the Gaussians which are used to control the cubic splines that define Class B

Table 2.1



The four control Gaussians and the track of $\underline{\mu}^{(t)}$*: t [0,1]*

Figure 2.3

The steps in generating a sample from Class B are given below. A parametric value, t, is chosen from a uniform distribution, U[0,1]. The three splines are evaluated at that t value, thereby defining an isotropic Gaussian distribution, G($\underline{\mu}^{(t)}$, $\underline{\underline{\Sigma}}^{(t)}$). A random sample is then generated from that distribution.

1) $t \in \mathbf{U}[0,1]$ where U[] denotes a uniform distribution

2) $$\underline{\mu}_0^{(t)} = \mathbf{BSpline}\left(t \,\middle|\, \underline{\mu}_{f_0}\left(G^{(0)}\right), \underline{\mu}_{f_0}\left(G^{(1)}\right), \underline{\mu}_{f_0}\left(G^{(2)}\right), \underline{\mu}_{f_0}\left(G^{(3)}\right) \right)$$

3) $$\underline{\mu}_1^{(t)} = \mathbf{BSpline}\left(t \,\middle|\, \underline{\mu}_{f_1}\left(G^{(0)}\right), \underline{\mu}_{f_1}\left(G^{(1)}\right), \underline{\mu}_{f_1}\left(G^{(2)}\right), \underline{\mu}_{f_1}\left(G^{(3)}\right) \right)$$

4) $\sigma^{(t)} = \mathbf{BSpline}\left(t \,\middle|\, \sigma\left(G^{(0)}\right), \sigma\left(G^{(1)}\right), \sigma\left(G^{(2)}\right), \sigma\left(G^{(3)}\right) \right)$ so $\underline{\underline{\Sigma}}^{(t)} = \begin{bmatrix} \left(\sigma^{(t)}\right)^2 & 0 \\ 0 & \left(\sigma^{(t)}\right)^2 \end{bmatrix}$

5) $\underline{x} \sim \mathbf{G}\left(\underline{\mu}^{(t)}, \underline{\underline{\Sigma}}^{(t)}\right)$

Class A represents the variety of other tissues present in an inhomogeneous MR image. It has been demonstrated that the inhomogeneities affect different tissues types differently [Aylward and Coggins 1994], and thus Class A extends throughout feature space and does not suffer consistently from the inhomogeneity. Class A is therefore represented by a multivariate Gaussian distribution with a large isotropic variance, i.e., its variance is circularly symmetric. Its parameters are given in Table 2.2.

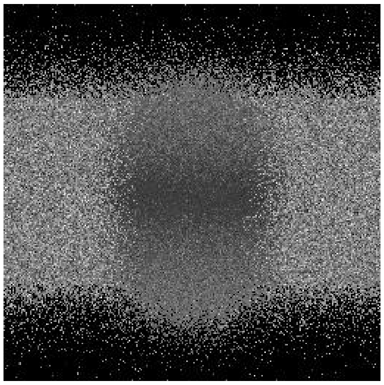
Class A	**f_0**	**f_1**
Mean	128	128
Covar	1296	0
f_0		
f_1	0	1296

The parameters of Class A
Table 2.2

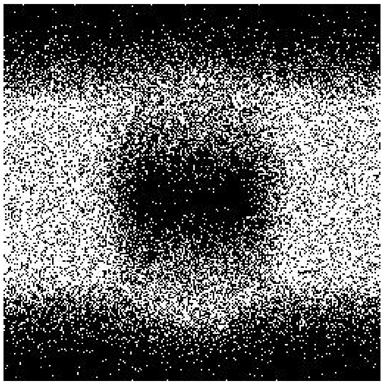
Training and Testing Set Sizes: Medium size tumors, multiple multiple sclerosis lesions, and other pathologies can cover regions in excess of 18mm^3. As a result, 900 samples, given the 1mm interslice and 5mm intraslice resolution commonly available using MR, are generally available from a single patient for training, i.e., $|S^{(tr:B)}|$=900. A total of 2700 samples are used for testing since it is also reasonable to expect that at least three other cases containing the tissue of interest would be available, i.e., $|S^{(te:B)}|$=2700. A variety of tools exist for the rapid extraction of these samples from the images. To indicate equal class *a priori* probabilities, the competing population, Class A, is represented by an equal number of training and testing samples, i.e., $|S^{(tr:A)}|$=900 and $|S^{(te:A)}|$=2700.

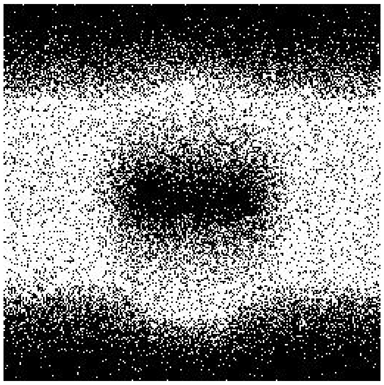
Other Inhomogeneous Modalities: Many other imaging modalities exhibit inhomogeneities. Inhomogeneities have been shown to be present in X-ray CT images as a result of beam hardening. Improper SPECT attenuation compensation can also demonstrate spatial correlations.

To demonstrate the correspondence between the Class A and Class B distributions and those present in inhomogeneous medical images, the distributions can be used to generate samples and produce a simulated a pseudo-medical image having two tissue types. Figure 2.4 is one possible pseudo-medical image based on these descriptions. Samples from Class A's distribution are uniformly spread across columns 0-64 and 196-256. Class B's samples are assigned to columns using a random pick from a Gaussian having a mean of 128 and a standard deviation of 32. The f_0 value generated for each sample specifies its row number, and the corresponding f_1 value specifies its intensity. One million samples were generated from each distribution to produce the image. Class B's intensity inhomogeneity is clearly visible as a correlation between its samples' intensity and row number. As a result of this inhomogeneity, it is difficult to distinguish samples by intensity near the top and bottom of this image. The effect of the inhomogeneity is displayed when an intensity thresholding is attempted to distinguish the classes. Two manual thresholdings further illustrate this effect (Figures 2.5 and 2.6).



Pseudo-medical image generated by interpreting f_0 *as image row and* f_1 *as image intensity.*
Class A samples occupy the right and left portions of the image.
Class B samples occupy the central track.
Figure 2.4





Two manual thresholdings of the pseudo-medical image in Figure 2.1.
The effect of the intensity/row inhomogeneity is clearly visible.
Figure 2.5 Figure 2.6

It can be imagined that such an image could result from SPECT. The central region, i.e., Class B, might correspond with damaged tissue having reduced uptake of the isotope. The inhomogeneity may have resulted from attenuation from intervening structures. To accurately estimate the extent of the damaged tissue, the inhomogeneity/attenuation must be compensated for. This dissertation presents a technique which is capable of accurately forming the necessary tissue models.

2.1.1.2. Problem 2 Description and Justification

Image processing techniques, i.e., mean field correction, can remove most of the effects of intensity inhomogeneities in many images [Aylward and Coggins 1994; Johnston, Atkins et al. 1996; Wells III, Grimson et al. 1996]. The second two-feature two-class problem arises from the elimination of the intensity inhomogeneities. The resulting populations are referred to as Class A′ and Class B′. Since the inhomogeneity correction has little effect on Class A, Class A and Class A′ have identical parameterizations. Class B, however, is transformed to an elliptical Gaussian distribution, Class B′. The parameters of Problem 2′s classes are given in Table 2.3.

Class A′		**f_0**	**f_1**
Mean		128	128
Covar		1296	0
f_0			
	f_1	0	1296

Class B′		**f_0**	**f_1**
Mean		128	56
Covar		576	0
f_0			
	f_1	0	324

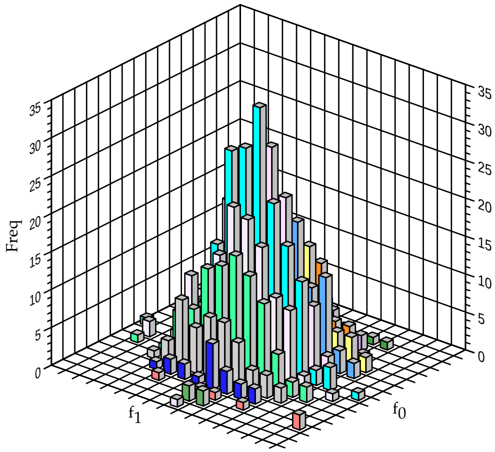
Parameters of Class A′ and Class B′
Table 2.3

2.1.2. Feature Space

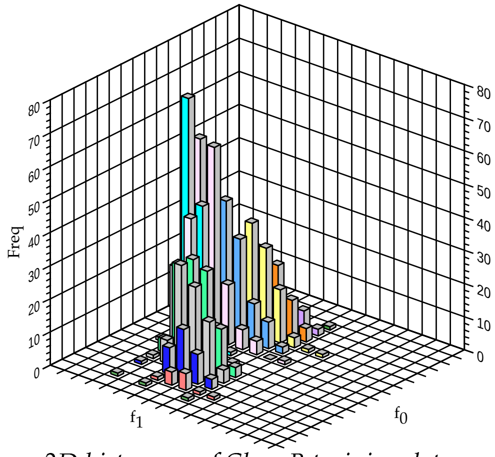
Feature space is the multidimensional range of the samples being considered. A sample consisting of N random variables, “features”, maps an object to a point in an N-dimensional “feature space.” For this dissertation all features are considered to be of commensurate units, and thus feature space is Euclidean. For most pattern recognition problems this is an acceptable assumption. When it does not hold, techniques exist for rescaling the feature values so that their marginal / individual variances are normalized and thus their units are made commensurate, i.e., factor analysis of correlation and covariance [Duda and Hart 1973; Jain and Dudes 1988; Jain 1989; Mao and Jain 1995]. Neural network researchers have demonstrated numerous applications in which such transforms have proven to be beneficial to the parameter selection processes [Mao and Jain 1995]. However, this normalization should be applied with care since it is possible to eliminate exactly the relationship sought in the data [Jain 1989].

2.1.3. Scattergrams

Consider the 2D histograms shown in Figures 2.7 and 2.8. While perhaps visually appealing, occlusion is a significant problem when attempting to inspect the distribution of the data when projected into 2D without user interaction.

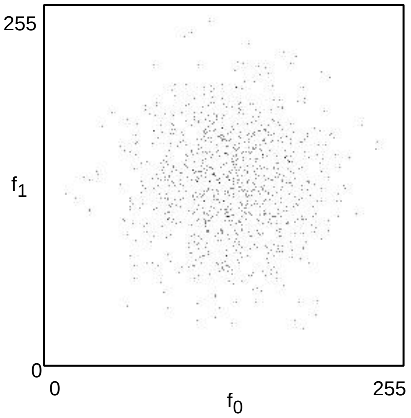


2D histogram of Class A training data
Figure 2.7

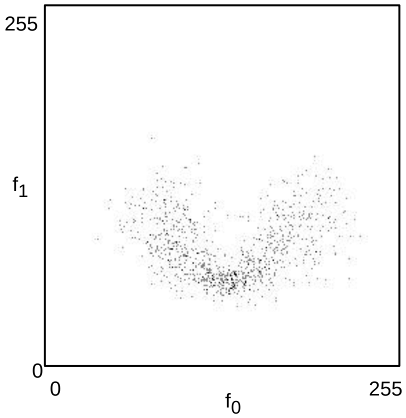


2D histogram of Class B training data
Figure 2.8

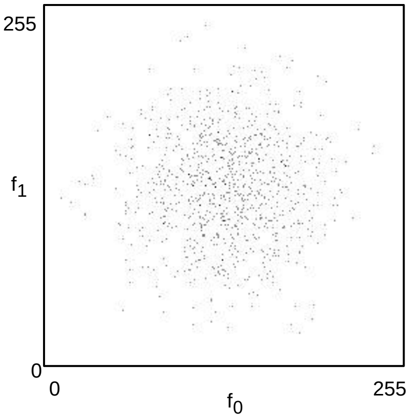
Scattergrams are 2D plots in which the intensity at each point corresponds to the relative



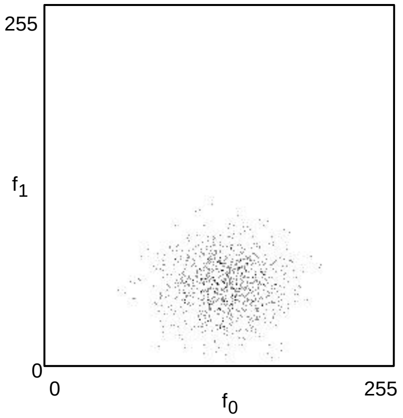
Scattergram of Class A,
a circularly symmetric Gaussian
Figure 2.9



Scattergram of Class B,
a generalized projective Gaussian
Figure 2.10



Scattergram of Class A′,
a circularly symmetric Gaussian,
with parameters identical to Class A
Figure 2.11



Scattergram of Class B′,
a elliptical Gaussian,
representing a corrected Class B
Figure 2.12

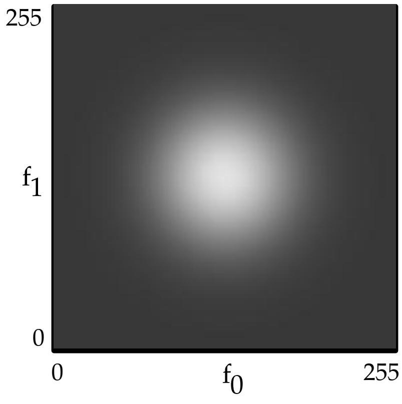
frequency of occurrence of a combination of two feature values in a collection of samples. The shape of the distribution of the samples is better revealed in this manner (Figures 2.9 through 2.12). Groups of samples having similar values produce clouds of high intensity.

If N>2, it is possible to develop scattergrams whose axes correspond to the projection of the samples onto a combination of features or hyperfeatures, i.e., a linear combination of features, e.g., eigenplots [Duda and Hart 1973; Jain 1989]. Scattergrams are not limited to coordinate axes; certain applications benefit from visualizing the data via scattergrams having polar axes.

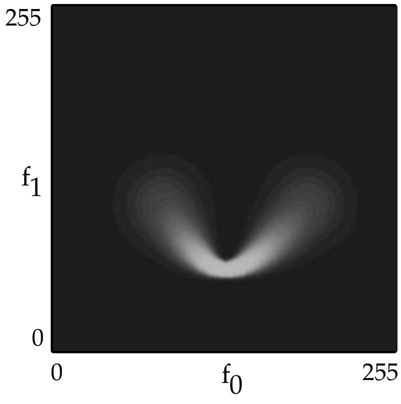
Given the digital domain in which the algorithms of this dissertation are being implemented, only discrete feature spaces and scattergrams are being considered. That is, feature values have a fixed level of precision, or, equivalently, they can only attain a finite set of values. This constraint is not limiting in most pattern recognition problems. Digital imaging techniques already exhibit these constraints. Binning is the process by which continuous data is mapped to a discrete format, i.e., a collection of bins / cells / buckets. A number of binning techniques exist which minimize the possibility that this transform will have a significant effect on the representation of the continuous distribution. This dissertation discusses binning in more detail in Section 4.3.

2.1.4. Population Distributions: Density Functions

Population distributions are represented using density functions. Given a sample, a density function approximates the probability of that sample having originated from the associated population. As a result, a density function's domain is feature space, and it has a unit integral. Density functions can be visualized in a scattergram to provide a qualitative understanding of their shape and overlap, i.e., consider Figures 2.13 and 2.14 in which brighter intensities correspond to higher probabilities.



Population Density of Class A
Figure 2.13



Population Density of Class B
Figure 2.14

Density functions can be defined explicitly or implicitly. Explicit representations estimate the parameters of the function from the training samples. For example, a Gaussian density function is completely specified by a mean and covariance. Using the samples of the population represented by the scattergrams in Figures 2.9 through 2.12, the means and covariances in Table 2.4 are produced.

The random effects inherent in a finite empirical sampling cause the estimated parameters to differ from one instance of a sampling to another. As expected, however, the estimated parameters (Table 2.4) and the underlying population parameters (Tables 2.1, 2.2, and 2.3) closely match for the Gaussian shaped populations, i.e., Class A, A′, and B′.

Class A	**f_0**	**f_1**
Mean	128.345	128.288
Covar	1247.353	0.055
f_0		
f_1	0.055	1335.568

Estimated Gaussian parameters of Class A

Class B	**f_0**	**f_1**
Mean	131.113	84.201
Covar	1480.644	0.237
f_0		
f_1	0.237	477.967

Estimated Gaussian parameters of Class B

Class A′	**f_0**	**f_1**
Mean	128.345	128.288
Covar	1247.353	0.055
f_0		
f_1	0.055	1335.568

Estimated Gaussian parameters of Class A′

Class B′	**f_0**	**f_1**
Mean	128.230	56.144
Covar	554.379	0.018
f_0		
f_1	0.018	333.920

Estimated Gaussian parameters of Class B′

Table 2.4

2.1.5. Decision Bounds and Sample Labeling

Decision bound specification is the deciding factor in the labeling accuracy of any classification system. Decision bounds exist as hypersurfaces in feature space. They delineate the regions, “decision regions”, in feature space within which all test samples will be assigned the same label. As with population density functions, decision bounds can be either explicitly or implicitly represented by a pattern recognition system.

The explicit representation of the decision bounds eliminates the need for the explicit representation of a density function. It follows that assumptions concerning the shapes of the decision bounds imply assumptions regarding the shapes of the distributions and vice versa. Consider the well known linear decision bound classification system. It makes the implicit assumption that the populations are well represented by linear combinations of the features and univariate Gaussians (Section 2.3.1).

When decision bounds are implicit, explicit density functions are used by the corresponding pattern recognition system. Labels are assigned using Bayesian decision theory.

The probability that a sample came from class C is computed based on the class conditional probability of that sample, $\mathbf{P}(\underline{x}\,|\,C)$, the class' *a priori* probability, $\mathbf{P}(C)$, and that sample value's probability, $\mathbf{P}(\underline{x})$:

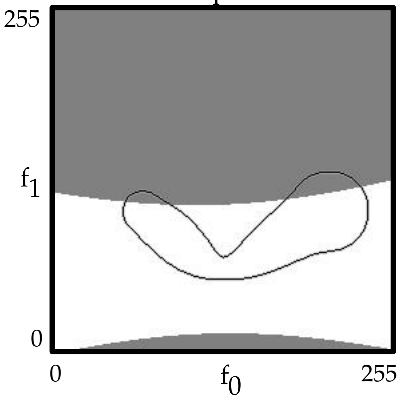
$$\mathbf{P}\left(C|\underline{x}\right)=\frac{\mathbf{P}(C)\,\mathbf{P}\left(\underline{x}|C\right)}{\mathbf{P}\left(\underline{x}\right)} \qquad [2.1]$$

The value of $\mathbf{P}(\underline{x}\,|\,C)$ is provided by the density function, and the value of $\mathbf{P}(C)$ is usually equal to the portion of training samples from Class C. A sample is assigned a population's label based on which class is the most likely to have generated that sample. When comparing $\mathbf{P}(C\,|\,\underline{x})$ across classes, the sample's prior probability, $\mathbf{P}(\underline{x})$, can be factored out of Equation 2.1. When the classes have equal priors, $\mathbf{P}(C)$, that value can also be eliminated. As a result, a sample, $\underline{x}$, can be assigned a label, $i=1..N_C$, via Equation 2.2:

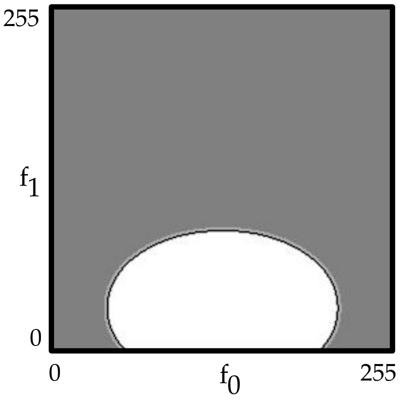
$$\underset{i\in[1..N_C]}{\mathbf{arg\,max}}\left(P\left(C^{(i)}\right)P\left(\underline{x}|C^{(i)}\right)\right) \qquad [2.2]$$

2.1.6. Labeling Feature Space

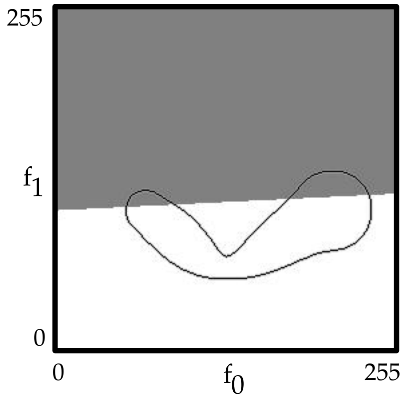
By evaluating every point in feature space and mapping each class label to a unique intensity, an image of the decision regions of feature space can be created. Figures 2.15 through 2.18 result from the application of Gaussian and linear classifiers, defined from the samples shown in the scattergrams in Figures 2.9 to 2.12, to every point in feature space. The optimal decision bounds for each problem are shown as black curves.



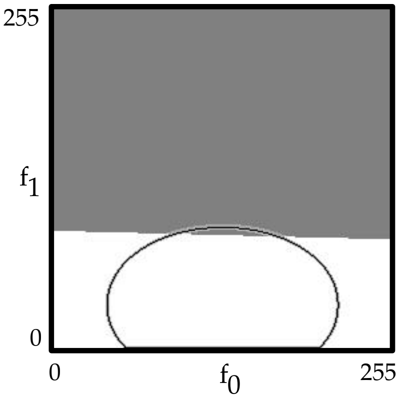
Class A / B decision regions produced by Gaussian classification with optimal decision bounds overlaid
Figure 2.15



Class A′ / B′ decision regions produced by Gaussian classification
Figure 2.16



Class A / B decision regions produced by linear classification
Figure 2.17



Class A′ / B′ decision regions produced by linear classification
Figure 2.18

2.2. Comparing Pattern Recognition Systems

The performance of a classifier can be quantified using several criteria: development memory requirements, development time, operating memory requirements, labeling speed, labeling accuracy, labeling consistency, and ease of qualitative and quantitative analysis.

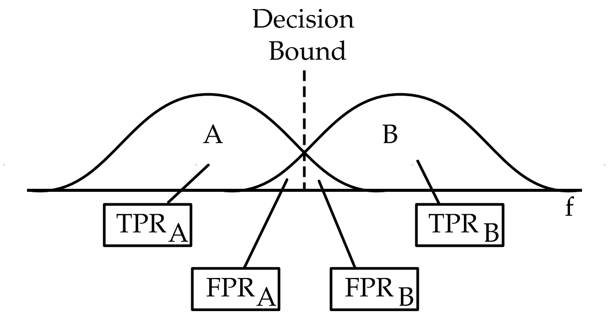
For the problems being addressed by this dissertation, classifier development memory requirements, development time, operating memory requirements, and labeling speed are not considered. For situations in which these factors are important, a different set of pattern recognition systems would need to be considered.

2.2.1. Labeling Accuracy

The accuracy of a classification technique must be judged by its performance on a specific problem. Performance is commonly quantified using true-positive and false-positive rates.

Because different classifiers make different distribution shape assumptions, the specification of a problem is important. If a classifier’s assumptions are correct for the chosen problem, that classifier will provide optimal accuracy limited only by the quality of the features. However, if for a different problem its assumptions are incorrect, extremely poor labelings can result.

The true-positive rate and the false-positive rate can be calculated for each population in the data. For these measures to be meaningful, it is important not to use the testing samples during the development of the classification system. That is, do not test on the training data. It is also important to know the correct labels of the testing samples.



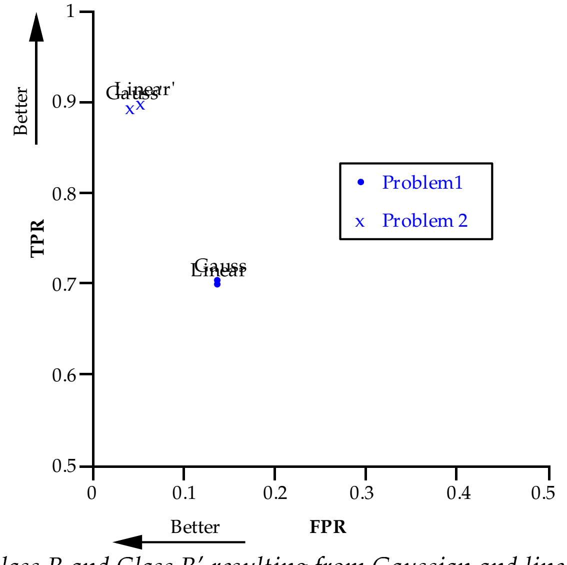
TP and FP regions of a single feature, two class problem having Gaussian population distributions.
Figure 2.19

The true-positive rate (TPR) of a population is the portion of test samples from that class which were assigned the correct label by the classification system. A class's false-positive rate (FPR) indicates the portion of test samples which were incorrectly assigned that class' label by the classification system (Figure 2.19).

For example, using the Gaussian and the linear classifiers depicted in Figures 2.15 through 2.18, and the 2700 testing samples from each class, the TPRs and FPRs shown in Table 2.5 result.

		TPR	**FPR**
Class B	**Gauss**	0.700	0.140
(versus Class A)	**Linear**	0.696	0.138
Class B′	**Gauss′**	0.891	0.045
(versus Class A′)	**Linear′**	0.896	0.056

TPRs and FPRs for linear and Gaussian classifiers
Table 2.5



FPR/TPR for Class B and Class B′ resulting from Gaussian and linear classification
Figure 2.20

These results can be visualized using plots of TPR-versus-FPR (see Figure 2.20). Each classifier's performance is summarized by a point on these plots. The symbol ′ is used to designate results from Problem 2, i.e., Class A′ versus Class B′.

The influence of the assumptions embodied in the classifier type is evident. Assuming Class B is Gaussian is incorrect, so both classifiers perform poorly in Problem 1. For Problem 2, however, the Gaussian assumptions are correct, and the Gaussian classifier produces excellent accuracy. Even though the density function assumptions of the linear classifier are not completely upheld (Section 2.3.1), the quality of the measures involved is such that the linear classifier is still able to produce accurate results.

2.2.2. Labeling Consistency: Monte Carlo One-Sigma of Labeling Accuracy

Classifier consistency will be quantified by the Monte Carlo one-sigma[1] range of the true positive and false positive rates[Sobol' 1994]. A one-sigma range for a measure is related to the standard error range of that measure, and as with standard error ranges, smaller one-sigma ranges correspond to reduce measure variability. That is, classifiers with small true positive and false positive one-sigma ranges are said to be more consistent.

Given a sufficient number of training samples and accurate distribution shape assumptions, the TPRs and FPRs associated with a pattern recognition system for a particular problem will be consistent despite the specific collection of training samples used. However, as the assumptions fail or as the training samples becomes sparse, the performance of a classification system can begin to vary as the collection of training samples changes.

Classifier performance can also vary because of multiple, non-optimal local extrema in the measure being optimized during the development of a classification development. This is usually the case when a classifier's parameter values are determined using an iterative technique or require the prior specification of a hyperparameter, e.g., the number of components. The initial values of the parameters may influence the final accuracy of the classifier as much as the collection of training samples used. Multilayered perceptrons trained via backpropagation, FGMMs trained via MLEM, and numerous other pattern recognition systems require such additional considerations. It will be shown that when MLEM is used to develop a FGMM, local maxima and non-optimal global maxima exist in the likelihood measure and as a function of their hyperparameter. These extrema can result in large variations in classifier accuracy dependent on

[1] "one-sigma" defines a measure of standard error; it will not be represented by a greek symbol. The greek symbol σ is reserved for measures of variance or scale.

the collection of training samples used and the starting point in the parameter space (Section 2.3.7). The existence of and difficulties associated with these extrema were a significant motivating factor for this dissertation.

Labeling consistency is measured via Monte Carlo simulation in which a classifier's TPRs and FPRs are recorded for different, yet constant in size, sets of training and testing samples.

Monte Carlo one-sigma values are proportional to standard error estimates. They specify the 67% confidence intervals for the values of interest, e.g., for the average true positive and false positive rates. If R Monte Carlo runs record an average TPR value of $\mu^{(TPR)}$ and a TPR standard deviation of $\sigma^{(TPR)}$, the Monte Carlo true positive one-sigma value is defined as

$$\text{one} - \text{sigma}^{(TPR)} = \frac{\sigma^{(TPR)}}{\sqrt{R}} \qquad [2.3]$$

and the 67% confidence interval for the true positive rate is thus

$$\mu^{(TPR)} - \frac{\text{one} - \text{sigma}^{(TPR)}}{\sqrt{R}} < \text{TPR} < \mu^{(TPR)} + \frac{\text{one} - \text{sigma}^{(TPR)}}{\sqrt{R}} \qquad [2.4]$$

To simultaneously capture the consistency of multiple measures, the covariance matrix of those measures is used. Specifically, their consistency is revealed by the one-sigma range of the square root of the determinant of that covariance matrix. A Monte Carlo one-sigma value can be calculated for the combined variance of the TPR and FPR rates. Their covariance matrix is $\underline{\underline{\Sigma}}^{(TF)}$ formed from the collection of TPR and FPR values recorded during the R Monte Carlo runs. The square root of the determinant of that matrix summarizes the covariance of those measures.

$$\sigma^{(TF)} = \left| \underline{\underline{\Sigma}}^{(TF)} \right|^{1/2} \qquad [2.5]$$

The Monte Carlo one-sigma value of the combined TPR/FPR variance is

$$\text{one} - \text{sigma}^{(TF)} = \frac{\sigma^{(TF)}}{\sqrt{R}} \qquad [2.6]$$

2.2.3. Ease of Qualitative and Quantitative Analysis

The insight gained through the analysis of the representations formed by a pattern recognition system is as important as the labelings they produce. Such analysis allows the questions listed at the beginning of Chapter 1 to be answered.

For example, if a population is known to have a Gaussian distribution, various methods exist for

1) identifying outlying samples
2) specifying confidence intervals for the estimated parameters based on the number of samples used
3) producing receiver-operator characteristic curves which define the progression of TPR-versus-FPR given different error costs

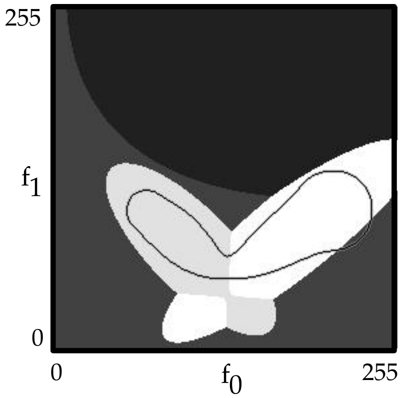
Additionally, the concepts of mean and covariance are simple enough to facilitate qualitative interpretation. These values can provide significant insight into the source of the populations and the nature / difficulty of the pattern recognition problem at hand.

2.3. A Comparison of Classification Systems

This section compares seven different classification techniques: linear, Gaussian, K nearest neighbor (KNN), Parzen windows (PW), multilayered perceptron (MLP), K means (KM), and finite Gaussian mixture modeling via maximum likelihood expectation maximization (FGMM). The presentation of each classifier is organized into three sections: operation, labeling accuracy and consistency, and ease of analysis.

Operation: These sections provide a high level description of each classification technique. No effort is made to provide the specific implementation details. A variety of books contain such information including: [Duda and Hart 1973; Jain 1989; Press, Flannery et al. 1990].

These sections also contain labeled scattergrams resulting from the application of the classification techniques to the training data shown in the scattergrams in Figures 2.9 to 2.12. Figure 2.21 is an example of one such labeling generated by a FGMM via MLEM classifier for Problem 1.



Example labeling of the Class A/B scattergram provided
by a FGMM via MLEM classifier with K=2 per class (FGMM02)
Figure 2.21

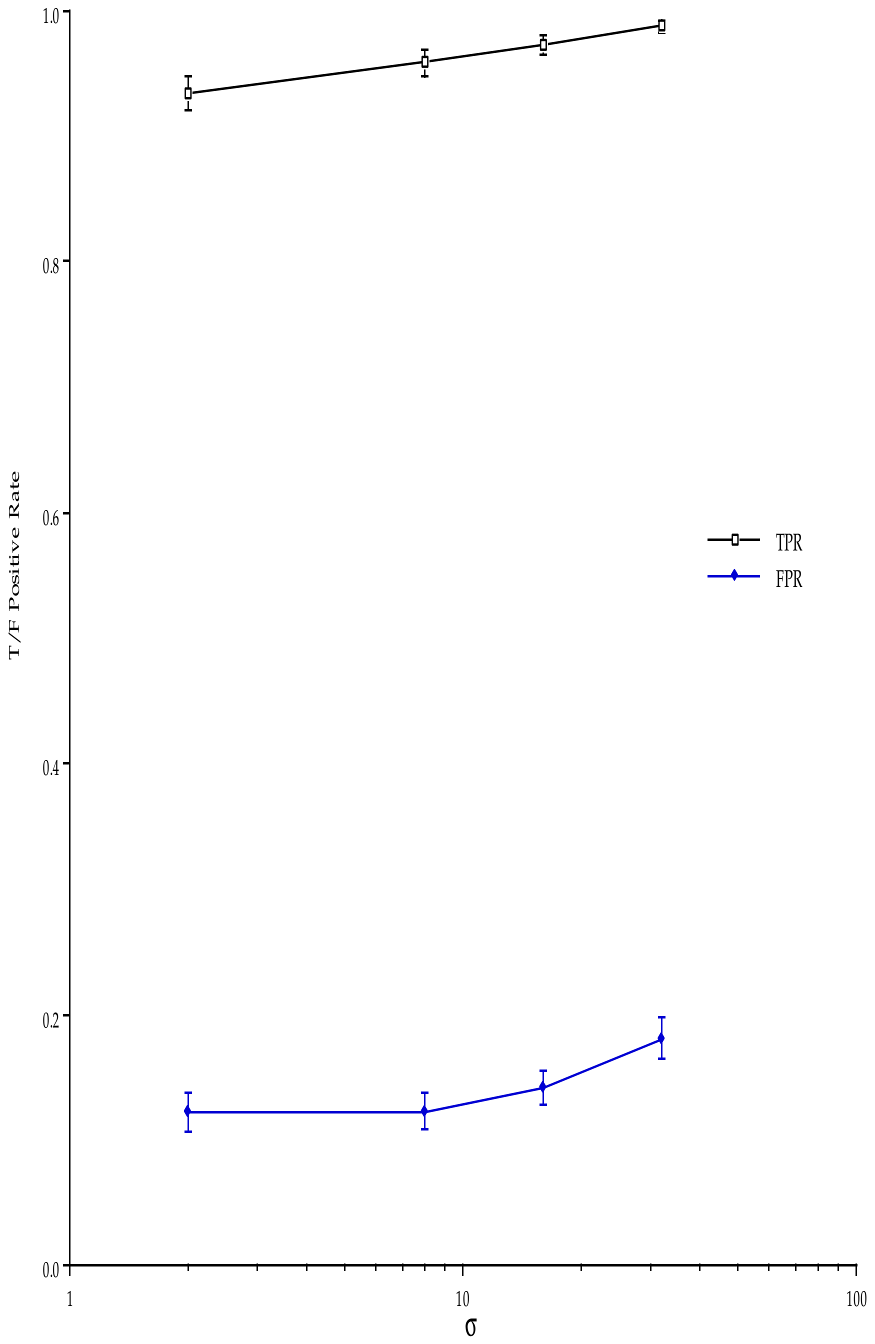
Dark gray regions are associated with Class A/A′ labelings. Regions labeled in light gray correspond to Class B/B′ samples. When multiple components are used to model a distribution, different shades of gray are used to distinguish their subregions. In Figure 2.21, there are two Gaussian components per class model as indicated. Overlaid onto these scattergrams are outlines of each problem's optimal decision bound.

Labeling Accuracy and Consistency: These sections provide the summary statistics from a Monte Carlo study involving each classifier's true-positive and false-positive rates on 1000 different collections of training and testing data from the classification problems presented in Section 2.2. When hyperparameters exist for a classification technique, e.g., the parameter K in K means, the classifier's performance for a variety of hyperparameter values is explored. The average and Monte Carlo one-sigma values of the Class B and Class B′ TPRs and FPRs are given in table form as in Table 2.6. Section 2.3.8 provides plots of Class B/B′ FPR-versus-TPR for all of the classifiers.

Method.	**TPR**	**FPR**	**TPR One-Sigma**	**FPR One-Sigma**	**Combined One-Sigma**
Gauss	0.86057	0.29396	0.01756	0.01737	0.01636

Example measures of accuracy and consistency for Gaussian classifier for Class B
Table 2.6

When appropriate these sections also provide graphs of a classifier's hyperparameter value versus its average FPR and TPR value. The Monte Carlo one-sigma ranges of these values are also indicated on the graphs using high and low markers. Figure 2.22 is an example of such a graph.



Example plot of FPR/TPR versus parameter σ for Parzen Windows for Class B'.
Includes Monte Carlo one-sigma range (standard error) for each rate.
Figure 2.22

Ease of Analysis: These sections discuss the qualitative and quantitative analysis of the representations formed by each classification system. The goal is to provide a rough assessment of how easy it is to interpret the representations provided by each method. References will be provided, and a few of the key strengths, weakness, and methods will be briefly mentioned.

2.3.1. Linear

Linear classifiers operate using linear, i.e., hyperplane, decision bounds. These classifiers make the assumption that the populations are well represented by univariate Gaussians applied to a weighted linear combination of the features.

Operation: The normal direction of the hyperplane is defined by the maximum eigenvalued eigenvector, i.e., Fisher's linear discriminate, of the Hotelling matrix of the training samples. This vector in feature space specifies the weighted linear combination of the features to which the univariate Gaussian is fit. This combination of features is said to define a "hyperfeature"/"latent feature" that best distinguishes the populations. The populations' means and variances along this normal direction define the univariate Gaussians and thereby position the oriented plane in feature space.

The Hotelling matrix is calculated using the class priors, **P**(A) and **P**(B), the class mean vectors, $\underline{\mu}^{(A)}$ and $\underline{\mu}^{(B)}$, and the class covariance matrices, $\underline{\underline{\bullet}}^{(A)}$ and $\underline{\underline{\bullet}}^{(B)}$. It is a multivariate signal-to-noise measure.

The global mean is calculated as

$$\underline{\mu} = \sum_{k\in\{A,B\}} \mathbf{P}(k)\underline{\mu}^{(k)} \qquad [2.7]$$

The signal matrix captures the spread of the means of the classes about the global mean:

$$\underline{\underline{S}}_{ij} = \sum_{k\in\{A,B\}} \mathbf{P}(k)\left(\underline{\mu}_i^{(k)} - \underline{\mu}_i\right)\left(\underline{\mu}_j^{(k)} - \underline{\mu}_j\right) \qquad [2.8]$$

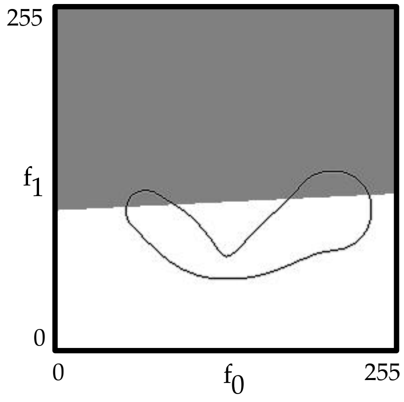
The noise matrix is the weighted sum of the spread of each class' data about their means:

$$\underline{\underline{N}}_{ij} = \sum_{k\in\{A,B\}} \mathbf{P}(k)\underline{\underline{\Sigma}}_{ij}^{(k)} \qquad [2.9]$$

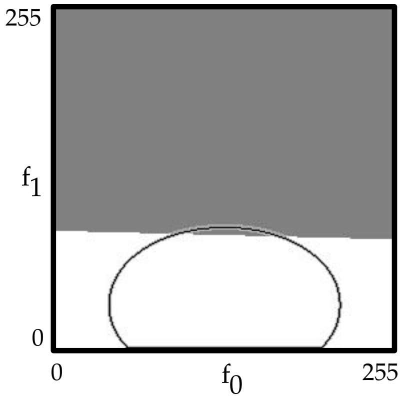
The Hotelling matrix, $\underline{\underline{H}}$, is the ratio of these two matrices.

$$\underline{\underline{H}} = \underline{\underline{N}}^{-1}\underline{\underline{S}} \qquad [2.10]$$

Figures 2.23 and 2.24 show the decision regions developed via linear classification given the data represented by the scattergrams shown in Figures 2.9 through 2.12.



Class A / B decision regions produced by Linear classification
Figure 2.23



Class A′/B′ decision regions produced by Linear classification
Figure 2.24

Labeling Accuracy and Consistency: When a linear classifier's assumptions are correct, its results are optimally accurate and consistent given the quality of the features being used. This is not the case for the two problems at hand. Tables 2.7 and 2.8 summarize the accuracy and consistency of the linear classifier for these problems.

Method.	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
Linear	0.86497	0.31235	0.06153	0.06942	0.03564

TPRs, FPRs, and consistency for Class B and from linear classification
Table 2.7

Method.	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
Linear	0.95270	0.12630	0.02965	0.03227	0.02061

TPRs, FPRs, and consistency for Class B′ and from linear classification
Table 2.8

Ease of Analysis: The eigenvector with the maximum eigenvalue from the Hotelling matrix is the direction of maximum separation of the classes. That vector defines a hyperfeature that consists of the weighted linear combination of the original features that best differentiates the populations. For example, when the features are generated via spatial filters, as is the case with multiscale offset Gaussians features [Coggins 1990; Coggins 1992; Coggins and Graves 1994], the weights can be used to combine the original spatial filters to specify a single filter, "hyperfilter" or "hyperfeature", which is tuned to differentiate the populations in the problem at hand. By generating such a filter, only that spatial filter needs to be applied to future images to collect the hyperfeature and distinguish the populations. By visualizing that filter, significant insight into the problem can be attained.

The weighting provided by the eigenvalue can also be used to specify an ordering with which to select the features and thereby reduce the dimensionality of feature space. Often 10-12 samples per independent feature are needed to generate an accurate multivariate Gaussian representation of a population [Neter, Wasserman et al. 1978]. If the number of samples available is limited, the weighting provides a useful ranking of the utility of the features.

2.3.2. Gaussian

Gaussian classifiers operate under the assumption that the populations are well represented by Gaussian shaped distributions. Gaussian classifiers form explicit Gaussian representations of the distributions and are one of the most popular techniques used in classification. Linear, elliptical, and hyperbolic decision bounds can be implicitly formed between two competing Gaussian distributions.

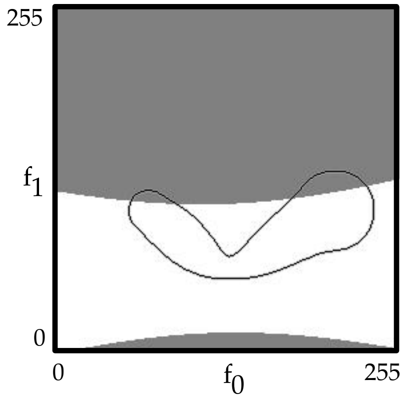
<u>Operation</u>: The mean and covariance matrix of a population specify its Gaussian representation. By reducing the equation of a Gaussian, the Bayesian classification process can be reformulated as a minimum distance process. With equal class priors, distance is judged using the Mahalanobis distance measure

$$\mathbf{D}_{M}\left(\underline{x};\underline{\mu},\underline{\underline{\Sigma}}\right)=\left(\underline{x}-\underline{\mu}\right)^{t}\underline{\underline{\Sigma}}^{-1}\left(\underline{x}-\underline{\mu}\right) \qquad [2.11]$$

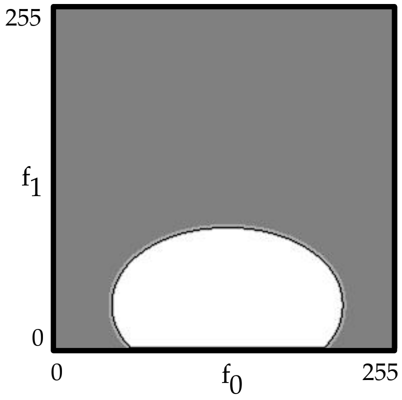
and samples are assigned the label of the class, i, to which they are “closest”:

$$\underset{i\in[1..N_{C}]}{\operatorname{argmin}}\left(\mathbf{D}_{M}\left(\underline{x};\mu^{(i)},\underline{\underline{\Sigma}}^{(i)}\right)\right) \qquad [2.12]$$

Figures 2.25 and 2.26 show the decision regions formed for the two pattern recognition problems using Gaussian classification.



Class A / B decision regions produced by Gaussian classification
Figure 2.25



Class A′/B′ decision regions produced by Gaussian classification
Figure 2.26

Labeling Accuracy and Consistency: While the Gaussian assumptions are incorrect for the first problem, they are correct for the second; note the drastic change in absolute and relative accuracy and consistency (Table 2.9 and Table 2.10).

As with other techniques, optimal accuracy results when the assumptions are upheld.

Method.	**TPR**	**FPR**	**TPR One-Sigma**	**FPR One-Sigma**	**Comb. One-Sigma**
Gauss	0.86057	0.29396	0.01756	0.01737	0.01636

TPRs, FPRs, and consistency for Class B from Gaussian classification
Table 2.9

Method.	**TPR**	**FPR**	**TPR One-Sigma**	**FPR One-Sigma**	**Comb. One-Sigma**
Gauss′	0.95211	0.11352	0.00966	0.01185	0.01013

TPRs, FPRs, and consistency for Class B′ from Gaussian classification
Table 2.10

Ease of Analysis: As a result of being one of the most popular classification techniques and its relatively few and clearly defined parameters, Gaussian classification is probably one of the best understood and most intuitively informative representation methods. Gaussian classification is the standard against which all other techniques are compared in regard to ease of qualitative and quantitative analysis.

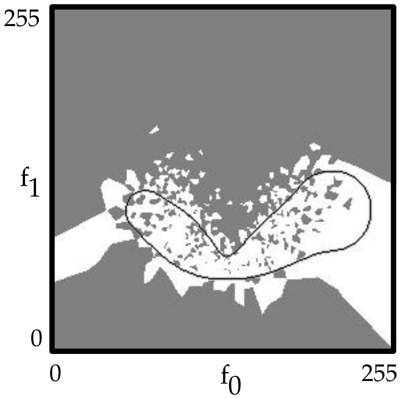
2.3.3. K Nearest Neighbor

If we allow the assignment of labels to be completely data driven, making no assumptions as to distribution parameters or shape, K nearest neighbor (KNN) classification results.

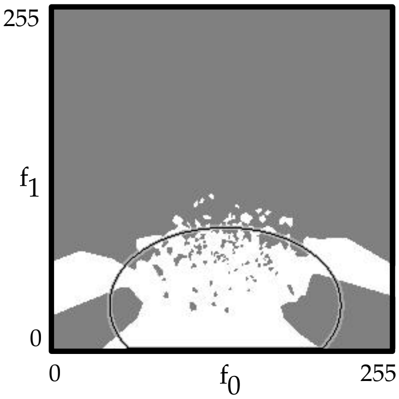
Operation: KNN classification assumes that the most common label among the K closest samples is the most probable label. Closeness is commonly judged using the Euclidean distance measure.

$$D_E(\underline{x},\underline{y}) = \left(\sum_{i=1}^{N} (\underline{x}_i - \underline{y}_i)^2 \right)^{1/2} \qquad [2.13]$$

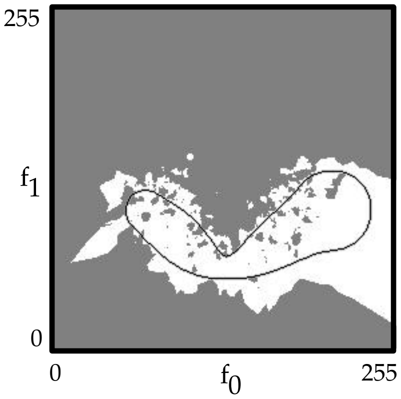
The results are given in Figures 2.27 to 2.34 for both of the pattern recognition problems for a variety of K values.



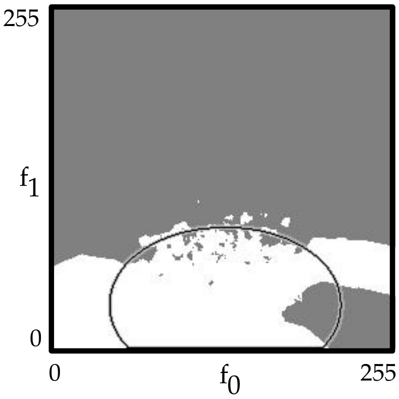
Class A / B decision regions produced by 1-nearest neighbor classification
Figure 2.27



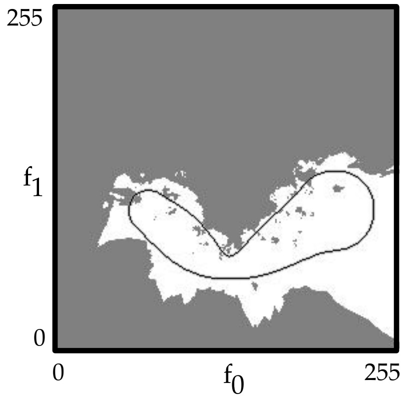
Class A′/B′ decision regions produced by 1-nearest neighbor classification
Figure 2.28



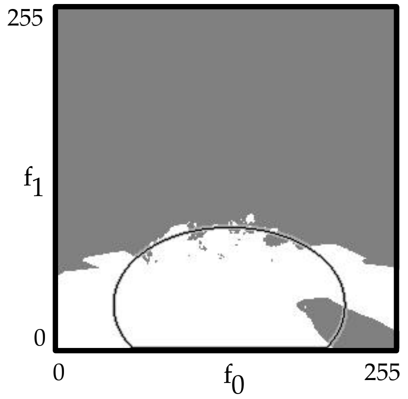
Class A / B decision regions produced by 3-nearest neighbor classification
Figure 2.29



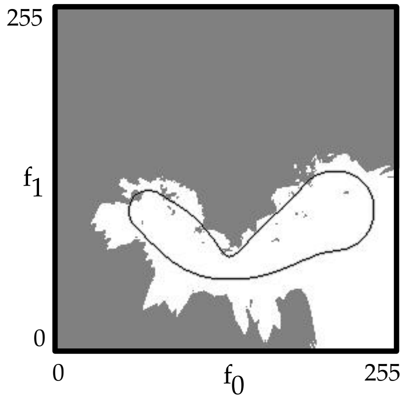
Class A′/B′ decision regions produced by 3-nearest neighbor classification
Figure 2.30



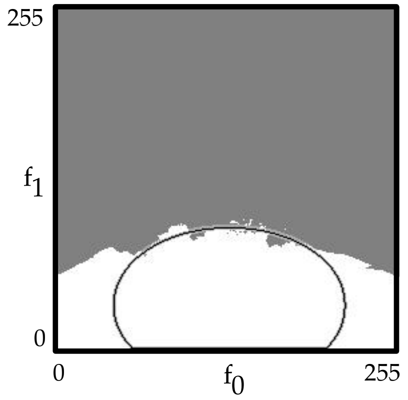
Class A / B decision regions produced by 7-nearest neighbor classification
Figure 2.31



Class A′/B′ decision regions produced by 7-nearest neighbor classification
Figure 2.32



Class A / B decision regions produced by 11-nearest neighbor classification
Figure 2.33



Class A′/B′ decision regions produced by 11-nearest neighbor classification
Figure 2.34

Labeling Accuracy and Consistency: The asymptotic total error rate, i.e., (1-TPR) + FPR, of K=1 nearest neighbor classification is at most twice the optimal Bayesian total error rate [Duda and Hart 1973; Schalkoff 1992].

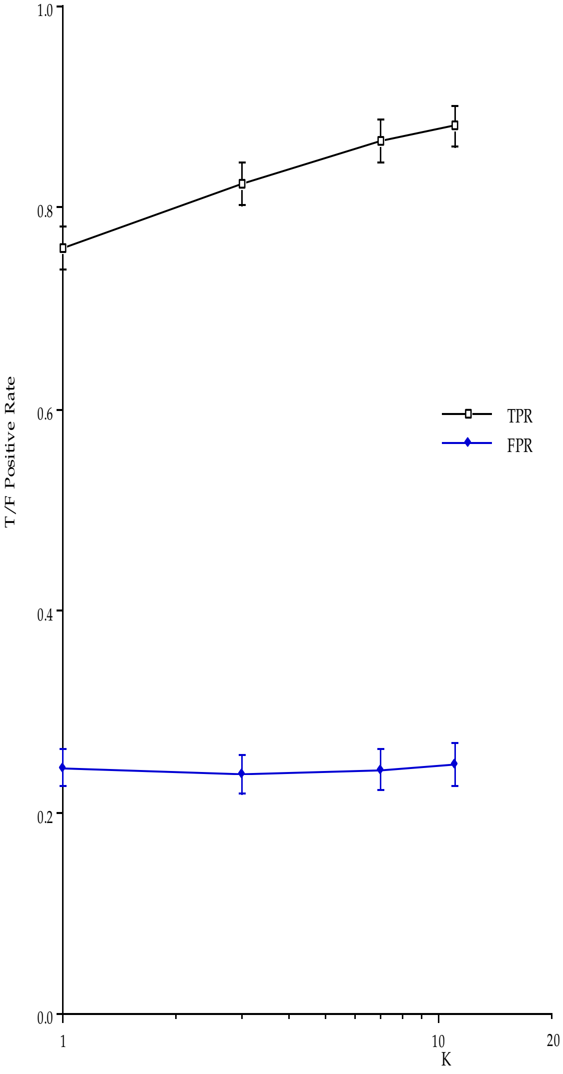
The most significant consideration for KNN classification is the value of K. The localized voting process of KNN makes it an approximate smoothing technique. As K increases, the region in feature space over which the labels are averaged increases. Too small of a K value results in undersmoothing the distribution of the labels (Figure 2.22). Too large of a K value produces oversmoothing (Figure 2.29). The optimal K value varies for each problem. Parzen windows (defined in Section 2.3.4) research indicates that the optimal neighborhood

Method.	K	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
KNN	1	0.76094	0.24433	0.02126	0.01861	0.01959
	3	0.82346	0.23834	0.02152	0.01895	0.01949
	7	0.86584	0.24230	0.02137	0.02049	0.01924
	11	0.88107	0.24680	0.02085	0.02146	0.01906

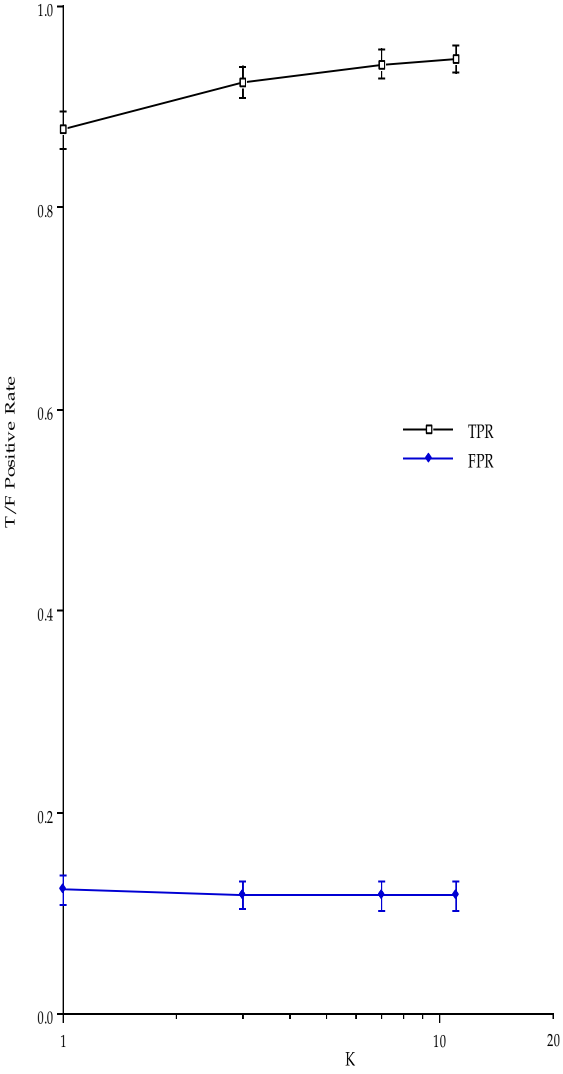
TPRs, FPRs, and consistency for Class B from K nearest neighbor classification
Table 2.11

Method.	K	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
KNN	1	0.87729	0.12305	0.01832	0.01420	0.01583
	3	0.92383	0.11823	0.01567	0.01395	0.01404
	7	0.94232	0.11722	0.01426	0.01441	0.01310
	11	0.94764	0.11751	0.01397	0.01481	0.01291

TPRs, FPRs, and consistency for Class B′ from K nearest neighbor classification
Table 2.12



K versus TPR/FPR for problem 1
Table 2.35



K versus TPR/FPR for problem 2
Table 2.36

size/scale and thus the optimal K value may even vary throughout feature space for a single problem. [Parzen 1962; Silverman 1978; Speckman 1988; Jones, Marron et al. 1994; Loader 1995; Babich and Camps 1996]

Ease of Analysis: KNN classification probably provides the least possible information in regard to the nature of the underlying population distribution. It provides no summary statistics and thus provides no means of qualitative or quantitative analysis. Furthermore, unlike Parzen windowing, it does not even provide a probability estimate on which subsequent processing can be performed.

2.3.4. Parzen Windows

Parzen windows (PW) is a kernel density estimation technique. Like KNN classification, it is considered a data driven technique. It is also considered a density interpolation technique because of its close relationship to convolution.

Its hyperparameters are the shape of the kernel and the size/scale/bandwidth of that kernel. It has been demonstrated that kernel scale, not kernel shape, is key to the accuracy and consistency of the representations formed by this technique [Silverman 1986]. As a result, the major area of research in regard to kernel density estimation concerns optimal scale/bandwidth estimation. Specifically, Parzen windowing has been extended to include functions which specify variations in kernel size based on the local distribution of the samples. Dependent on the error measure, e.g., mean integrated squared error, used to quantify the difference between the estimated and ideal density, automated, principled methods for variable kernel scale specification have been developed [Silverman 1978; Jones, Marron et al. 1994].

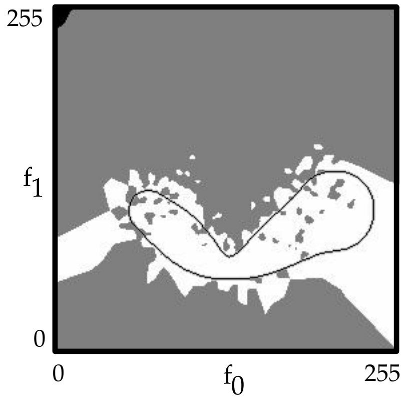
Operation: Instead of using a triangle or step function as Parzen did [Parzen 1962], this dissertation uses a Gaussian-shaped, fixed-scale kernel.

$$G(\underline{z},\sigma)=\frac{1}{(2\pi)^{N/2}|\sigma|^{N}}e^{-\frac{\underline{z}^{t}\underline{z}}{2\sigma^{2}}} \qquad \text{given } \underline{z}\in\Re^{N} \qquad [2.15]$$

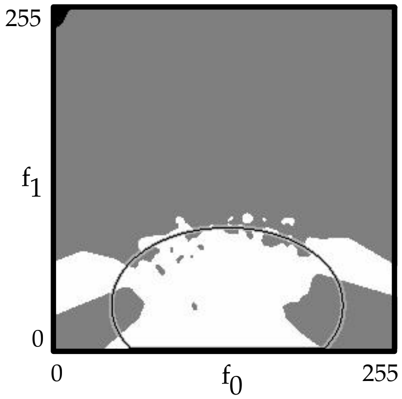
Given a set of samples *S*, the Parzen window density function is

$$\mathbf{P}(\underline{x},\sigma)=\frac{1}{|S|}\sum_{j=1}^{|S|}G\left(\underline{x}-\underline{y}^{(j)},\sigma\right) \qquad [2.16]$$

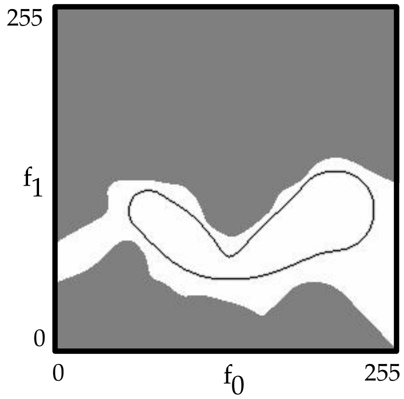
The results are given in Figures 2.37-2.44 for a variety of kernel scales. Figures 2.37 and 2.38 also contain regions labeled with black to indicate that no label could be assigned to those samples. The implementation used in this study evaluated only training samples within a fixed distance, 3σ, of each testing sample. When no training samples were within that distance, no density was estimated, and a black class label was assigned. These regions are a failure of the implementation and not the technique itself.



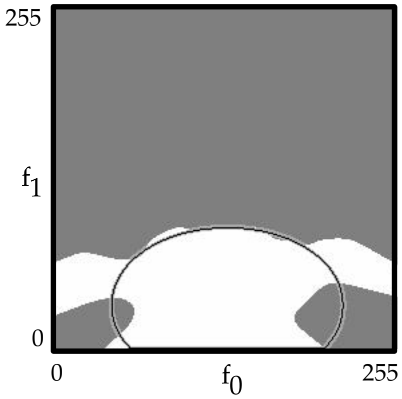
Class A / B decision regions produced by Parzen window, σ=2, classification
Figure 2.37



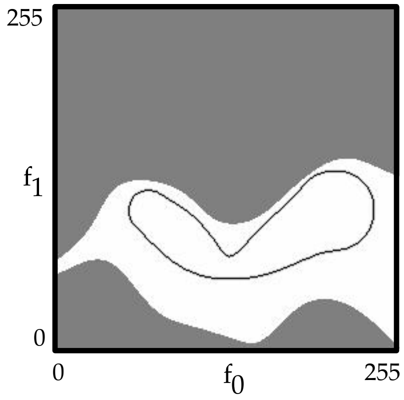
Class A′/B′ decision regions produced by Parzen window, σ=2, classification
Figure 2.38



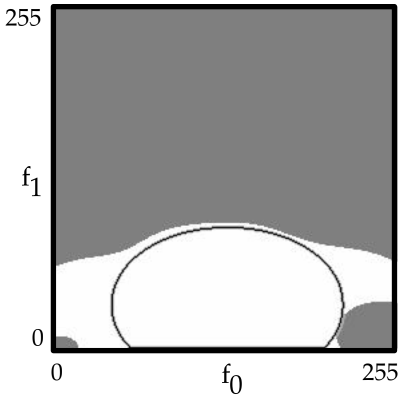
Class A / B decision regions produced by Parzen window, σ=8, classification
Figure 2.39



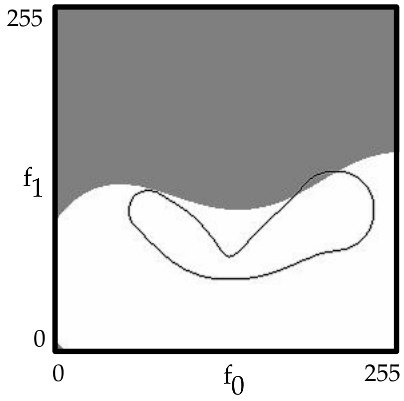
Class A′/B′ decision regions produced by Parzen window, σ=8, classification
Figure 2.40



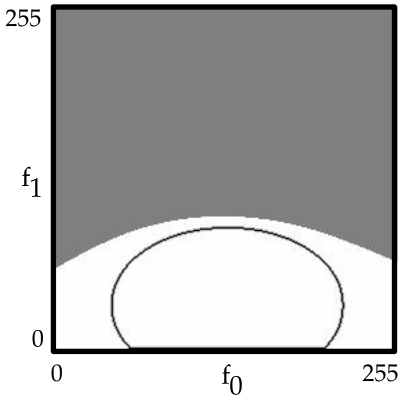
Class A / B decision regions produced by Parzen window, σ=16, classification
Figure 2.41



Class A′/B′ decision regions produced by Parzen window, σ=16, classification
Figure 2.42



Class A / B decision regions produced by Parzen window, σ=32, classification
Figure 2.43



Class A′/B′ decision regions produced by Parzen window, σ=32, classification
Figure 2.44

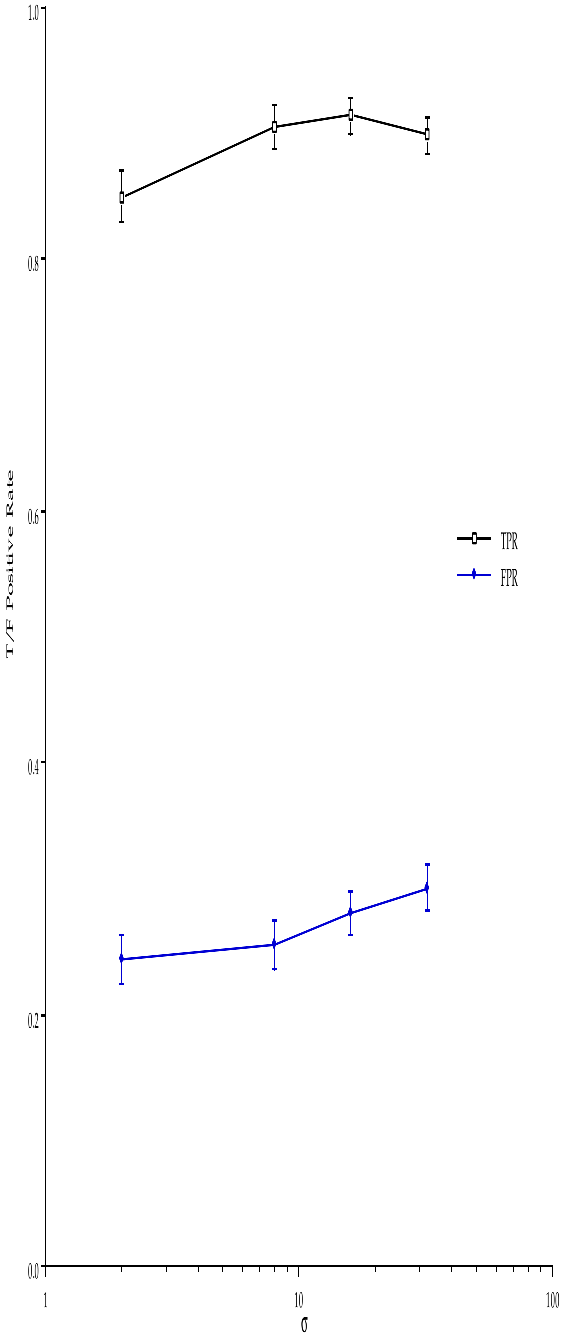
Labeling Accuracy and Consistency: Kernel density estimation techniques generally provide excellent accuracy and consistency for a range of kernel shapes and sizes. Significant research has gone into deriving functions for optimal kernel size specification [Silverman 1978; Speckman 1988; Jones, Marron et al. 1994; Loader 1995; Babich and Camps 1996]. As with KNN classification, using too small of a neighborhood/scale/bandwidth results in undersmoothing (Figure 2.37). Too large of a neighborhood produces oversmoothing (Figure 2.44).

Method.	**σ**	**TPR**	**FPR**	**TPR One-Sigma**	**FPR One-Sigma**	**Comb. One-Sigma**
PW	2	0.84960	0.24390	0.02069	0.01960	0.01911
	8	0.90534	0.25473	0.01700	0.01962	0.01650
	16	0.91394	0.28018	0.01458	0.01801	0.01515
	32	0.89889	0.30073	0.01420	0.01750	0.01504

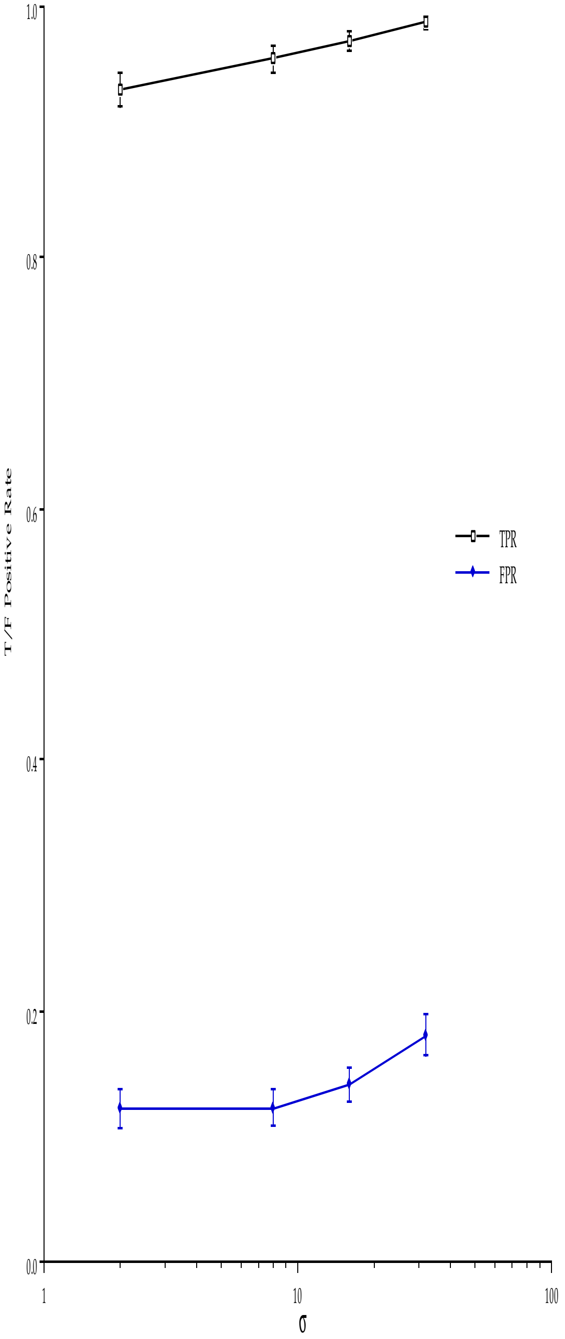
TPRs, FPRs, and consistency for Class B from Parzen Window classification
Table 2.13

Method.	σ	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
PW	2	0.93420	0.12209	0.01428	0.01476	0.01377
	8	0.95858	0.12271	0.01032	0.01375	0.01089
	16	0.97261	0.14125	0.00728	0.01416	0.00953
	32	0.98743	0.18084	0.00448	0.01625	0.00816

TPRs, FPRs, and consistency for Class B′ from Parzen Window classification
Table 2.14



TPR/FPR/Consistency versus σ for Class B
Figure 2.45



TPR/FPR/Consistency versus σ for Class B′
Figure 2.46

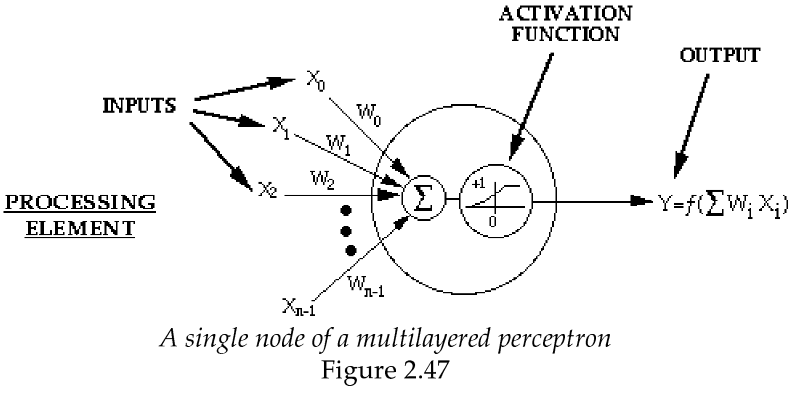
Ease of Analysis: In isolation, kernel density estimation techniques provide little qualitative or quantitative insight into the problems at hand. However, the resulting interpolated/smoothed density surfaces enable the statistical analysis of the data. For example, mode identification, ridge traversal, valley traversal, and numerous other techniques can be applied in a straightforward manner [Silverman 1978; Touzani and Postaire 1988; Lecocq and Postaire 1991; Cheng 1995].

2.3.5. Multilayered Perceptron

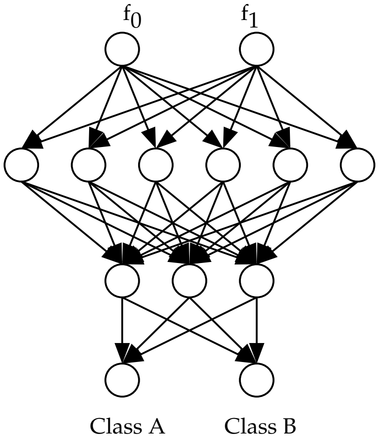
Multilayered perceptron neural networks (MLPs) are commonly called backpropagation networks. Backpropagation refers to the average root mean squared error (ARMSE) gradient descent parameter estimation technique, which is often applied to this style of feedforward network. [Lippmann 1987; Bebis and Georgiopoulos 1994]

MLPs develop decision bound representations. The complexity of the representations they form are dependent on the network architecture, the gradient descent step size, and the training time. Each of these considerations has been the subject of various gradient, genetic algorithm, simulated annealing, and heuristic strategies in an effort to automate the application of MLPs. [Schalkoff 1992; Bebis and Georgiopoulos 1994; Peterson, St. Clair et al. 1995] Of special interest is the modification to the network algorithm which results in the definition of radial basis function networks. These networks are closely related to finite Gaussian mixture models [Xu, Krzyzak et al. 1994].

This dissertation uses MLPs having two hidden layers and full feedforward connectivity between the layers (Figures 2.47 and 2.48). A network having 6 nodes in the first hidden layer and 3 nodes in the second hidden layer is references as MLP6x3.



A single node of a multilayered perceptron
Figure 2.47

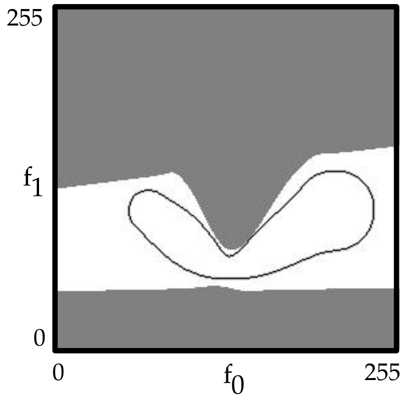


Perceptron with two hidden layered (MLP6x3)
Figure 2.48

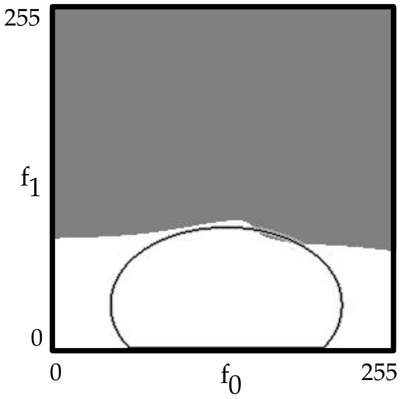
Operation: Each input node corresponds to a different feature. Each output node corresponds to a different class. A test sample is assigned the label associated with the output node which produces the largest output.

For Problems 1 and 2 training consists of 3,600,000 iterations, i.e., 667 passes through the training data. The weights are updated after each sample is presented, a strategy called "iterative" training. Using a root mean squared error measure, the weights are updated by taking a step in the gradient direction a distance of 1% of the gradient magnitude.

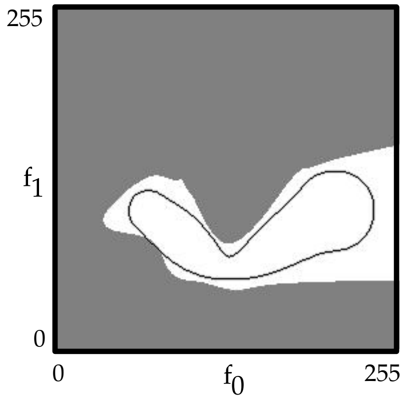
While not a criterion for the comparison of the classifiers in this dissertation, it is important to note that the training times associated with traditional backpropagation can be excessive. The time required was such that only R=100 runs of the Monte Carlo simulation could be performed in a reasonable amount of time. The Monte Carlo one-sigma values have been adjusted accordingly. The results for a variety of different sizes of two hidden layered MLPs are shown in Figures 2.49 to 2.54.



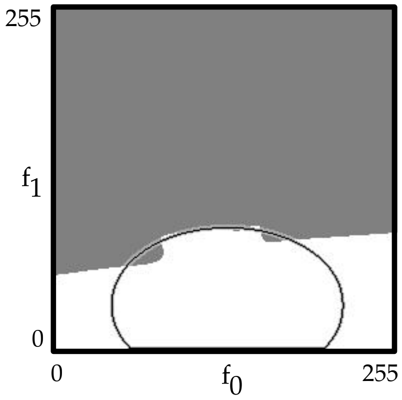
Class A / B decision regions produced by MLP, 6x3, classification
Figure 2.49



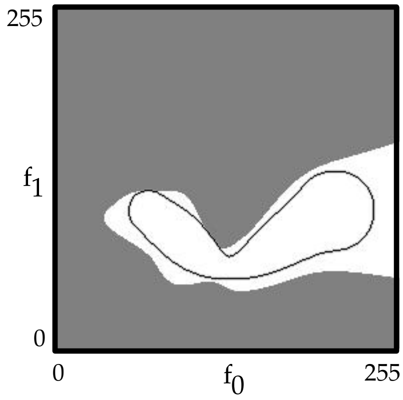
Class A'/B' decision regions produced by MLP, 6x3, classification
Figure 2.50



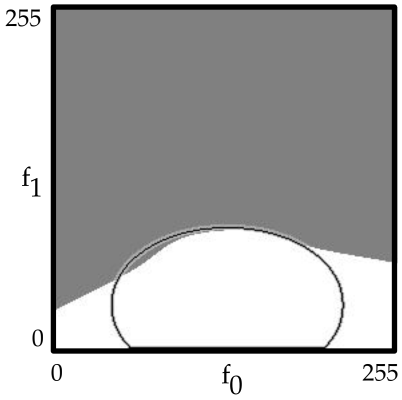
Class A / B decision regions produced by MLP, 12x6, classification
Figure 2.51



Class A'/B' decision regions produced by MLP, 12x6, classification
Figure 2.52



Class A / B decision regions produced by MLP, 24x12, classification
Figure 2.53



Class A'/B' decision regions produced by MLP, 24x12, classification
Figure 2.54

Labeling Accuracy and Consistency: The Stone-Weierstrauss theorem has been used to prove that MLPs can represent any function to an arbitrary degree of accuracy using one hidden layer , and by using two hidden layers, any decision bound can be represented [Hornik, Stinchcombe et al. 1989; Poggio and Girosi 1990; Osman and Fahmy 1994; Chen and Chen 1995]. The Monte Carlo results for Problems 1 and 2 are given in Tables 2.15 and 2.16. These results are plotted in Figures 2.55 and 2.56.

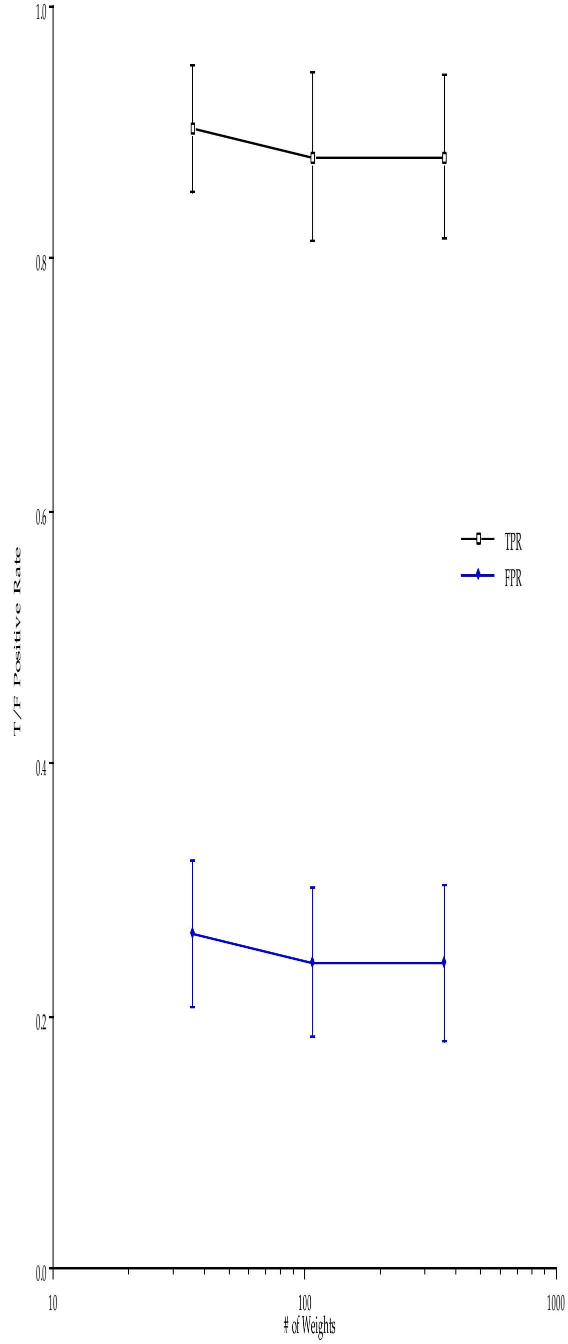
Method.	**#Wgts**	**TPR**	**FPR**	**TPR One-Sigma**	**FPR One-Sigma**	**Comb. One-Sigma**
MLP	36	0.90327	0.26495	0.05007	0.05858	0.03610
	108	0.88089	0.24267	0.06650	0.05939	0.04470
	360	0.88044	0.24143	0.06509	0.06155	0.04021

TPRs, FPRs, and consistency for Class B from MLP classification
Table 2.15

Method.	#Wgts	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
MLP	36	0.94569	0.11549	0.03472	0.03211	0.02013
	108	0.93519	0.10756	0.04273	0.03246	0.02343
	360	0.93376	0.10504	0.03667	0.02764	0.02150

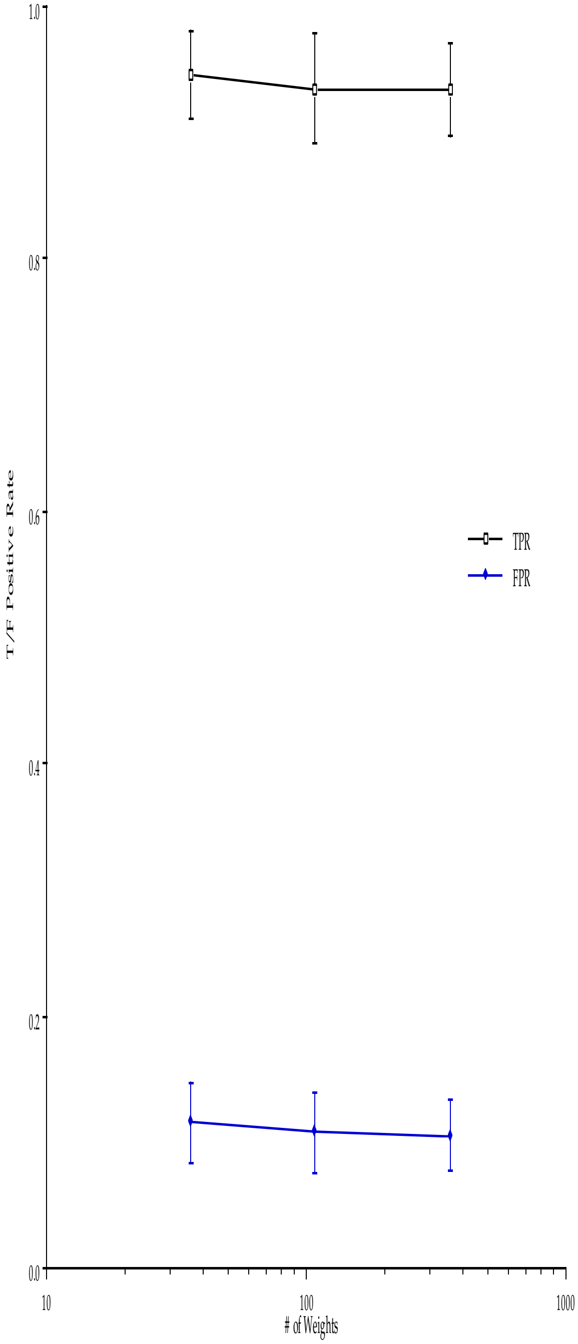
TPRs, FPRs, and consistency for Class B′ from MLP classification

Table 2.16



TPR/FPR/Consistency versus network size for Class B

Figure 2.55



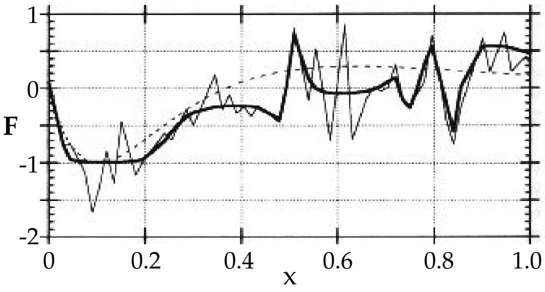
TPR/FPR/Consistency versus network size for Class B

Figure 2.56

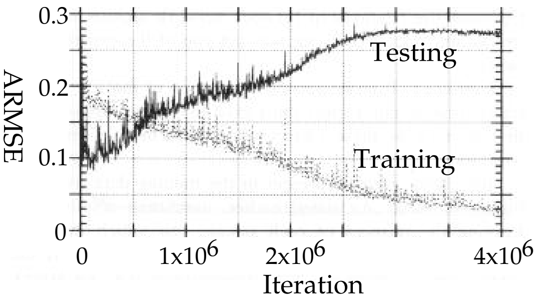
One of the problems with MLPs is overtraining. Consider the following function:

$$\mathbf{F}(x) = 4.26\left(e^{-x} - 4e^{-2x} + 3e^{-3x}\right) + \mathbf{G}(0,\sigma) \qquad \text{where } x = [0..3] \qquad [2.17]$$

G(0,σ) represents Gaussian additive noise with a zero mean. The σ=0 function is shown as the dotted line in Figure 2.57. The thin solid line in Figure 2.57 corresponds to the σ=0.1 function. When the latter is given to a MLP with one hidden layer of 30 nodes and trained for $4\text{x}10^6$ iterations, overtraining occurs. The function approximation developed by the MLP is shown as the thick line in Figure 2.57. The network has begun to fit to the noise of the samples. Figure 2.58 provides a plot of ARMSE for the training and testing data as the training of the network progresses. Overtraining is indicated by a rise in testing ARMSE despite the continual decrease in training ARMSE. Overtraining can also occur when a FGMM is given too many components to represent a distribution or when a polynomial of too high of a degree is used to represent a sampled function. With MLPs, however, controlling such error is not straightforward [Peterson, St. Clair et al. 1995].



Underlying function (dotted),
noisy sampling (thin line),
and MLP generated model (thick line).
Overtraining fits noise
Figure 2.57



Plot of training data's ARMSE
and testing data's ARMSE
versus training time.
Overtraining increases testing ARMSE
Figure 2.58

Ease of Analysis: One of the most common complaints concerning MLPs is that they are often viewed as black box solutions to problems. They provide little intuitive insight into the criterion with which they are making their decisions. Significant research has gone into developing algorithms for the quantitative analysis of their performance as well as providing a more intuitive representation of their decision process. One of the most common techniques relies on the conversion of the network to a decision tree or table lookup process. Further qualitative and quantitative analysis can then be performed on that structure. These approaches, however, do not completely address the validation and verification process required of black box methods.

2.3.6. K-Means

K-means (KM) operates under the assumption that the distribution of a collection of samples can be well represented by multiple circularly symmetric Gaussians having equal variance. As a result, K-means representations can be considered finite Gaussian mixture models having constrained Gaussian components. They implicitly form piecewise linear decision bounds.

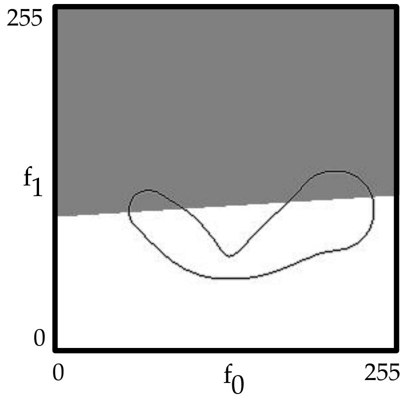
K-means can be applied as part of a classification scheme so as to model the distribution of a single population, or it can be applied to the training samples of multiple populations as a clusterer with the goal of automatically and efficiently distributing the means among the populations.

In this dissertation K-means is being used for classification. That is, it is being applied to each population independently to model the distribution of its samples and in turn provide a class conditional probability for Bayesian classification. Therefore, K=1 refers to modeling each population using one component and K=2 corresponds with two components per population and so forth.

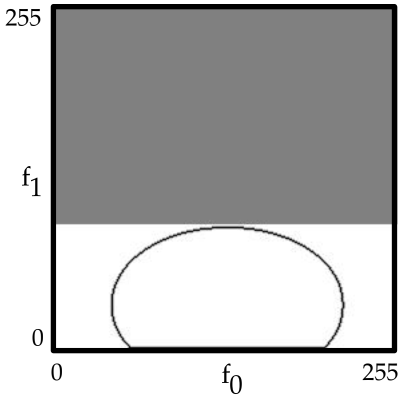
Operation: The application of K-means involves the following steps

```
1) Choose the number of components, K
For each population, C  [1..$N_c$], having a set of training samples, $S^{(tr:C)}$
    2) Choose $\underline{\mu}^{(C:\,i)}$ for all i=1..K
    3) Assign each $\underline{x}$  $S^{(tr:C)}$ to a component using minimum Euclidean distance criterion
    4) Recompute $\underline{\mu}^{(C:\,i)}$ for all i=1..K using component groupings from 3
    5) If every sample's component assignment is unchanged then end else goto step 3
```

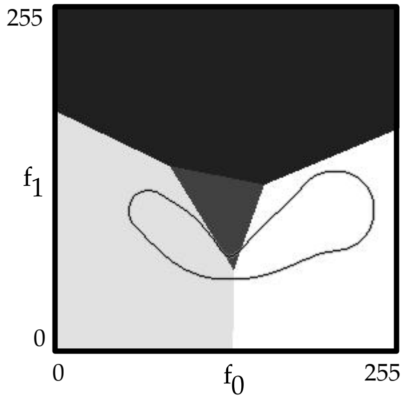
A variety of heuristics exist for choosing the initial means for the classes (the $\underline{\mu}^{(C:\,i)}$ in step 2). For the work presented in this dissertation, the $\underline{\mu}^{(C:\,i)}$ were chosen using K random selections from the training set $S^{(tr:C)}$.



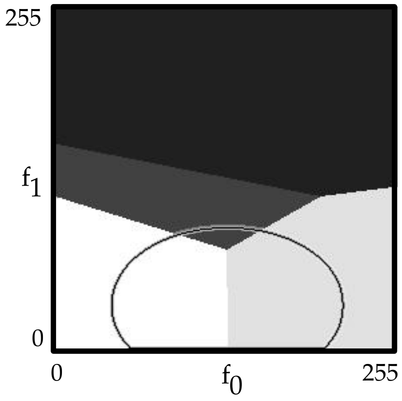
Class A / B decision regions produced by K=1 means classification
Figure 2.59



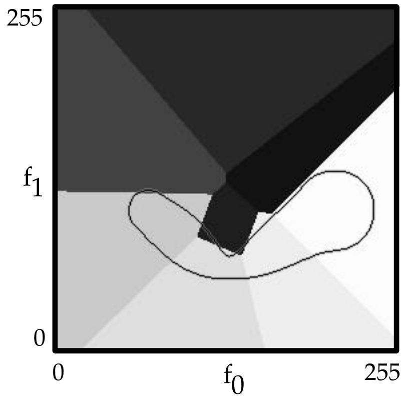
Class A′/B′ decision regions produced by K=1 means classification
Figure 2.60



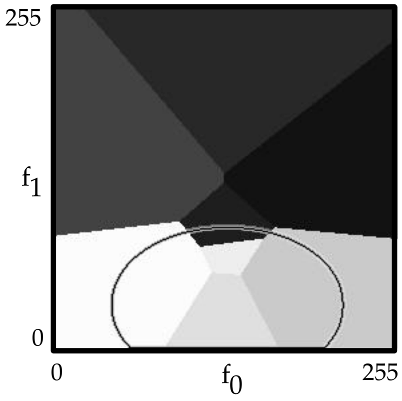
Class A / B decision regions produced by K=2 means classification
Figure 2.61



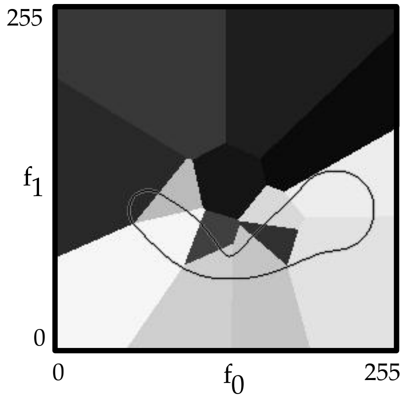
Class A′/B′ decision regions produced by K=2 means classification
Figure 2.62



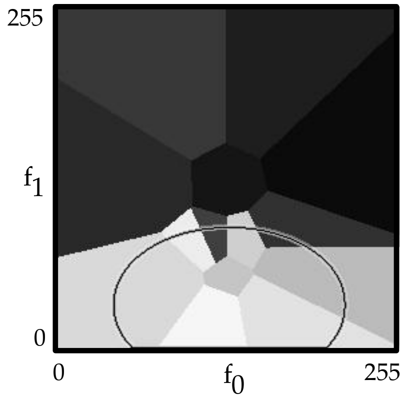
Class A / B decision regions produced by K=4 means classification
Figure 2.63



Class A′/B′ decision regions produced by K=4 means classification
Figure 2.64



Class A / B decision regions produced by K=7 means classification
Figure 2.65



Class A′/B′ decision regions produced by K=7 means classification
Figure 2.66

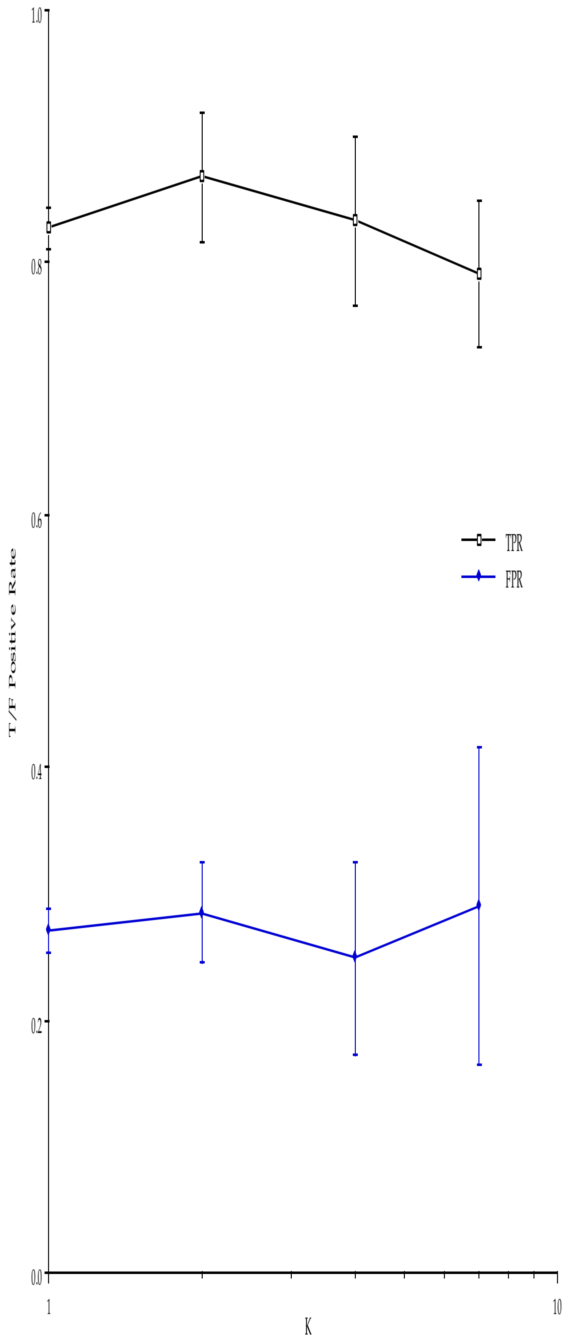
<u>Labeling Accuracy and Consistency</u>: The principal problems with K-means are its dependence on the initial mean values, its dependence on the order in which the training data is presented, its dependence on the specific set of training data being used, and the reliance on the user to specify K. That is to say, K-means is subject to local maxima and therefore provides poor labeling consistency. This is well illustrated in the graphs in Figure 2.67. For Problem 1, the drastic change in the one-sigma ranges for the TPR and FPR values as K increases indicates a severe decrease in labeling consistency. For Problem 1, ideally K=1 for Class A, but since Class B is an extruded Gaussian, it is difficult to determine an appropriate finite K value *a priori*. For Problem 2, K is known to be 1 for both classes. For both problems, the additional resources provided by larger K values appear to confound the solution. The problems with parameter initialization, user specification of K, and iterative parameter optimization techniques are studied more closely for FGMMs via MLEM in Section 2.3.7.

Method.	K	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
KM	**1**	0.82714	0.27073	0.01705	0.01739	0.01615
	2	0.86754	0.28483	0.05093	0.03924	0.03897
	4	0.83296	0.24890	0.06654	0.07657	0.07088
	7	0.79113	0.29009	0.05728	0.12592	0.08039

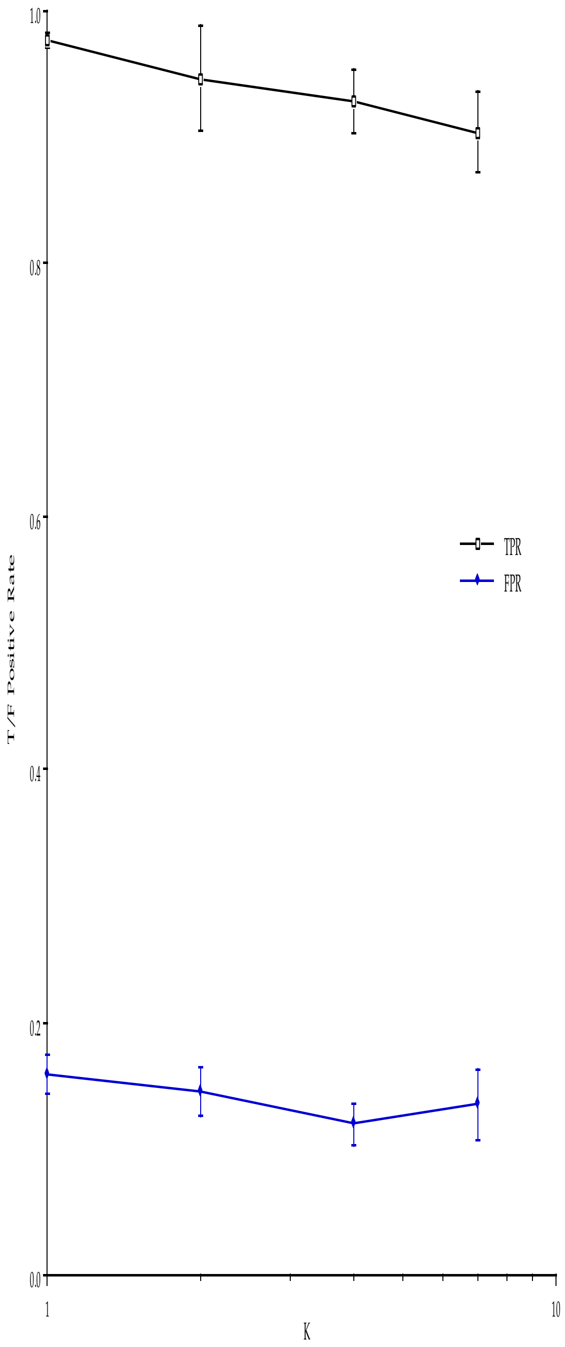
TPRs, FPRs, and consistency for Class B from K-means classification
Table 2.17

Method.	K	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
KM	**1**	0.97706	0.15863	0.00650	0.01508	0.00930
	2	0.94669	0.14549	0.04098	0.01911	0.02298
	4	0.92937	0.11924	0.02515	0.01614	0.01958
	7	0.90416	0.13444	0.03270	0.02753	0.02937

TPRs, FPRs, and consistency for Class B from K-means classification
Table 2.18

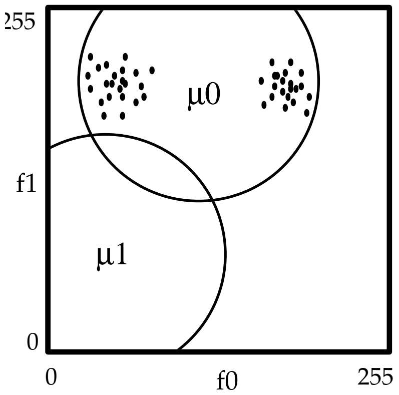


TPR/FPR/Consistency versus K for Class B
Figure 2.67



TPR/FPR/Consistency versus K for Class B'
Figure 2.68

Ease of Analysis: The use of K means to represent a population's distribution can provide significant qualitative and quantitative insight. The theorems of statistical analysis for Gaussian, i.e., K=1, and FGMM, i.e., K>1, also apply to K means. However, the existence of local maxima degrades the possibility of significant qualitative insight unless the quantitative analysis is also performed.



Local maximum for K=2 means
Figure 2.69

Consider the hypothetical training data and the (K=2)-means description of its distribution as shown via the means and isoprobability curves in Figure 2.69. Although a local maximum has been achieved, the performance is far from that of the ideal representation. Poor accuracy results, and any qualitative analysis of the means without a quantitative analysis would be misleading.

2.3.7. Finite Gaussian Mixture Modeling

Finite Gaussian mixture modeling develops representations of complex distributions through the weighted linear combination of multiple Gaussian component distributions. The probability of a sample $\underline{x}$ arising from a population which is represented by the FGMM parameterized by Ψ is given by

$$\mathbf{P}\left(\underline{x} \mid \Psi\right) = \sum_{i=1}^{K} \omega^{(i)}\,\mathbf{F}\left(\underline{x}, \Phi^{(i)}\right) \qquad \Psi = \left\{ \{\omega, \Phi\}^{(i)} \mid i = 1..K \right\} \qquad [2.18]$$

In 1886, Newcomb wrote the seminal paper on finite mixture modeling [Newcomb 1886]. He used a mixture of two univariate normals to model a distribution and its outliers. Pearson in 1894 presented a method of moments for automatically decomposing a mixture of normals [Pearson 1894]. His approach, however, required the solution of a ninth degree polynomial equation. Cohen in 1967 limited the problem to components with equal variances and presented a solution involving root finding and the first four moments [Cohen 1967]. Little attention was given to the likelihood maximization approach until 1972, when Tan and Chwang [Tan and Chang 1972], and independently Fryer and Robertson [Fryer and Robertson 1972] demonstrated that the method of moments is inferior to likelihood estimation for Cohen's limited problem and the more general cases.

In 1977, Dempster, Laird, and Rubin [Dempster, Laird et al. 1977] presented an iterative scheme for handling missing data in maximization problems and established its theoretical convergence properties. This iterative scheme is called expectation maximization (EM). It has been and continues to be applied to likelihood maximization for the definition of finite mixture models.

A variety of alternate technologies have also received considerable attention for FGMM development: graphical methods, minimum distance techniques such as chi-squared and least squared minimization methods, Bayesian techniques, Newton-Raphson, and the method of scoring. Although no single method has been shown to be ideal for all situations, EM has several properties which make it an appealing technique: [Jordan and Xu 1993; Xu and Jordan 1995; Zhuang, Huang et al. 1996]

1) EM is simple to apply: no matrix inversion is required as with Newton-Raphson and the method of scoring.

2) EM is stable: Bayesian techniques develop numerical difficulties in high dimensional parameter spaces.

3) EM converges to singularities less often

4) EM is monotonic: it is the only method for which likelihood is guaranteed to increase after each iteration

One of the most common complaints in regard to EM is its slow convergence rate. Recently, however, Xu [Xu and Jordan 1995] has shown that although EM strictly has first order convergence properties, the matrix which characterizes the direction of its step with respect to the local gradient serves to reduce the condition number of the Hessian of the maximum likelihood surface so that nearly second order convergence rates are often achieved.

2.3.7.1 Maximum Likelihood Estimation

The standard likelihood equation for a collection of training samples, *S*, given a GMM(Ψ) is

$$\mathbf{L}(S \mid \psi) = \prod_{j=1}^{|S|} \mathbf{P}\left(\underline{x}^{(j)} \mid \psi\right) \qquad [2.19]$$

Equation 2.19 is maximized (or minimized) when

$$\frac{\partial \mathbf{L}(S \mid \psi)}{\partial \psi} = 0 \qquad [2.20]$$

Finding a maximum likelihood estimate of *Ψ* is equivalent to finding the maximum log likelihood estimate of *Ψ:*

$$\mathbf{L}_{\mathrm{L}}(S \mid \psi) = \sum_{j=1}^{|S|} \mathbf{log}\left(\mathbf{P}\left(\underline{x}^{(j)} \mid \psi\right)\right) \qquad [2.21]$$

By assuming that each sample, $\underline{x}^{(j)}$, arises from one of the component distributions, $\mathbf{F}(\underline{x}^{(j)};\phi^{(i)})$, which exist in the mixture model in the portions $\omega^{(i)}$, then a new variable $z^{(ji)}$ can be introduced that captures the component membership for $\underline{x}^{(j)}$

$$z^{(ji)} = \begin{cases} 1 & \underline{x}^{(j)} \in \text{component i} \\ 0 & \text{otherwise} \end{cases} \qquad [2.22]$$

As a result, the log likelihood equation can be rewritten as

$$\mathbf{L}_{\mathrm{L}}(S \mid \psi) = \sum_{j=1}^{|S|} \sum_{i=1}^{K} z^{(ji)} \left(\mathbf{log}\left(\omega^{(i)}\right) + \mathbf{log}\left(\mathbf{F}\left(\underline{x}^{(j)}, \Phi^{(i)}\right)\right)\right) \qquad [2.23]$$

and this equation is maximal when

$$\frac{\partial \mathbf{L}_{\mathrm{L}}(S \mid \psi)}{\partial \psi} = \sum_{j=1}^{|S|} \sum_{i=1}^{K} z^{(ji)} \frac{\partial\, \mathbf{log}\left(\omega^{(i)}\right)}{\partial \omega} + \sum_{j=1}^{|S|} \sum_{i=1}^{K} z^{(ji)} \frac{\partial\, \mathbf{log}\left(\mathbf{F}\left(\underline{x}^{(j)}, \Phi^{(i)}\right)\right)}{\partial \phi} = 0 \qquad [2.24]$$

This formulation of the likelihood equation requires the specification of the $z^{(ji)}$, but during the development of the model, these terms are unknown. Thus the use of this equation as the function to be solved converts the maximum likelihood task to a "missing data" problem, which expectation maximization was developed to solve.

2.3.7.2 Maximum Likelihood Expectation Maximization

The EM algorithm was presented by Dempster, Laird, and Rubin [Dempster, Laird et al. 1977] as an iterative maximization algorithm capable of handling missing data. The EM algorithm is applied to FGMMs by treating all $z^{(ji)}$ as missing data. Each iteration of EM has two steps: an E-step and an M-step. The E-step, the expectation step, assigns values to the missing data variables based on the current model parameters. The M-step, the maximization step, adjusts the parameters of the model using the current estimates of the missing data variables.

The E-Step requires the calculation of expected value of the log likelihood function conditional on the initially estimated parameters, $\mathit{\Psi}^{(0)}$, so

$$\mathbf{Q}\left(\mathit{\Psi},\mathit{\Psi}^{(0)}\right)=\boldsymbol{E}\left\{\mathbf{L}_{\mathrm{L}}\left(S\middle|\mathit{\Psi}\right)\middle|\mathit{\Psi}^{(0)}\right\} \qquad [2.25]$$

This is achieved by substituting the component conditional posterior probabilities of each sample for the component membership variables, i.e., the missing data. The component conditional posterior probabilities, $\mathbf{P}^{(ji)}$, of each sample, j, for each component, i, are

$$z^{(ji)}\approx\mathbf{P}^{(ji)}=\mathbf{P}\left(\underline{x}^{(j)}\middle|\mathit{\Psi};i\right)=\frac{\omega^{(i)}\mathbf{F}\left(\underline{x}^{(j)};\phi^{(i)}\right)}{\sum\limits_{k=1}^{K}\omega^{(k)}\mathrm{F}\left(\underline{x}^{(j)};\phi^{(k)}\right)} \qquad [2.26]$$

The M-Step adjusts the estimates of the parameters $\mathit{\Psi}$ based on this new log likelihood estimate. The weights are estimated by

$$\omega^{(i)}=\frac{1}{|S|}\sum_{j=1}^{|S|}\mathbf{P}^{(ji)} \qquad \forall i=1..K \qquad [2.27]$$

and the remaining parameters of the model in the maximum log likelihood equation are determined by substituting of $\mathbf{P}^{(ji)}$ for $z^{(ji)}$ into the second term of Equation 2.23.

$$\sum_{j=1}^{|S|}\sum_{i=1}^{K}z^{(ji)}\frac{\partial\,\mathbf{log}\left(\mathbf{F}\left(\underline{x}^{(j)},\mathit{\Phi}^{(i)}\right)\right)}{\partial\mathit{\Phi}}=0 \qquad [2.28]$$

For a variety of component distribution shapes, solutions exist for this equation. Such is the case for FGMMs. Specifically, the mean and covariance of each component, i=1..K, are

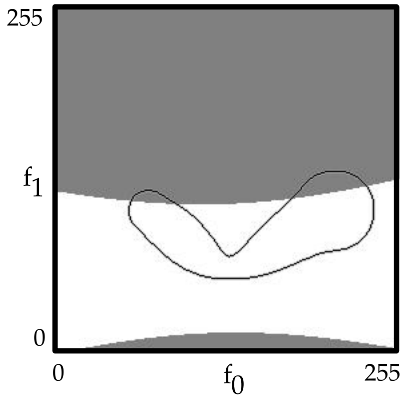
$$\underline{\mu}^{(i)}=\frac{1}{|S|}\sum_{j=1}^{|S|}\frac{\mathrm{P}^{(ji)}\underline{x}^{(j)}}{\omega^{(i)}} \qquad [2.29]$$

$$\underline{\underline{\Sigma}}^{(i)} = \frac{1}{|S|}\sum_{j=1}^{|S|} \frac{\mathbf{P}^{(ji)}\left(\underline{x}^{(j)}-\underline{\mu}^{(i)}\right)\left(\underline{x}^{(j)}-\underline{\mu}^{(i)}\right)^{t}}{\omega^{(i)}} \qquad [2.30]$$

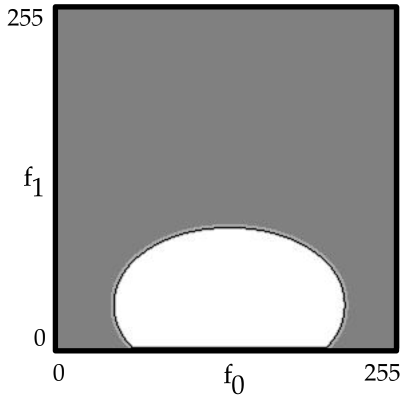
2.3.7.3 Operation

As with K means, FGMMs can be used as classifiers or clusterers. In this dissertation, as with K means, FGMMs are used to model the populations independently and provide class conditional probabilities to a Bayesian classifier.

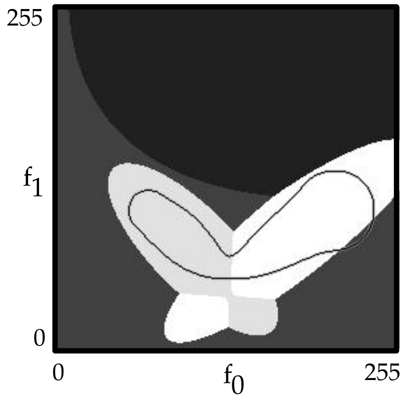
K means was used to initialize the model's parameters. The EM algorithm was run to convergence. Convergence was indicated by less than 0.01% change in the likelihood over two iterations. Analysis of the training indicates that convergence was reached in most cases in less than 10 iterations, but occasionally 250 were required.



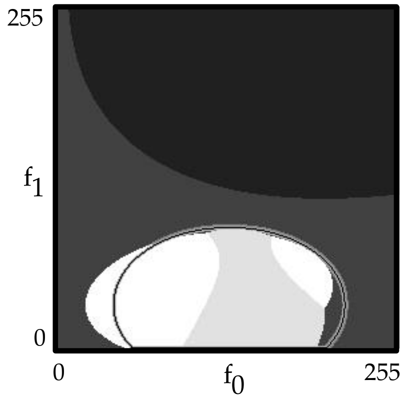
Class A / B decision regions produced by GMM via MLEM, K=1, classification
Figure 2.70



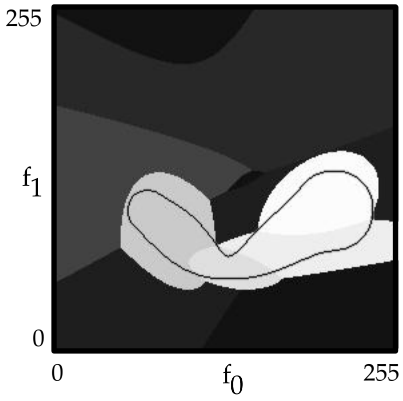
Class A'/B' decision regions produced by GMM via MLEM, K=1, classification
Figure 2.71



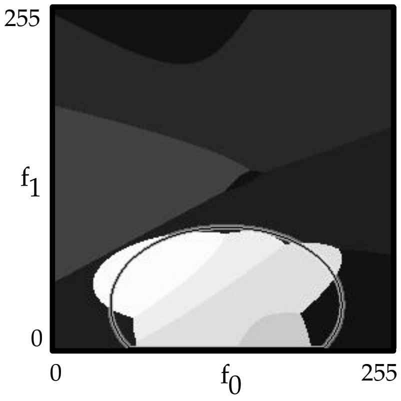
Class A / B decision regions produced by GMM via MLEM, K=2, classification
Figure 2.72



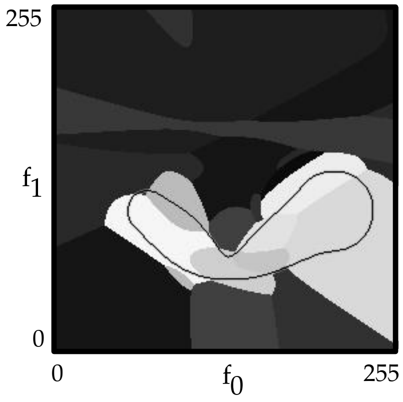
Class A'/B' decision regions produced by GMM via MLEM, K=2, classification
Figure 2.73



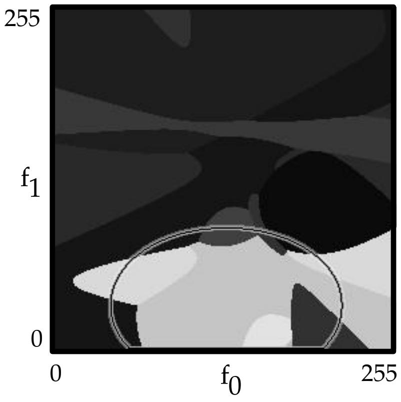
Class A / B decision regions produced by GMM via MLEM, K=4, classification
Figure 2.74



Class A′/B′ decision regions produced by GMM via MLEM, K=4, classification
Figure 2.75



Class A / B decision regions produced by GMM via MLEM, K=7, classification
Figure 2.76



Class A′/B′ decision regions produced by GMM via MLEM, K=7, classification
Figure 2.77

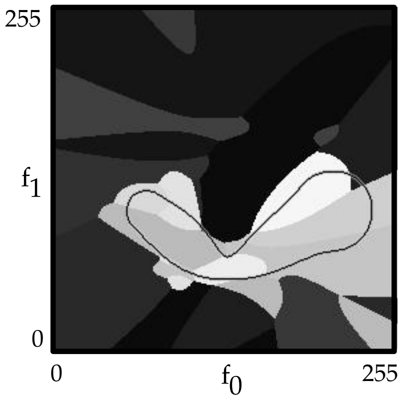
2.3.7.4 Labeling Accuracy and Consistency

It has been shown that FGMMs can be used to approximate arbitrary densities [Ferguson 1983]. In practice, however, FGMMs via MLEM are reliant on the user specification of K and are subject to local maxima and non-optimal global maxima.

A variety of methods have been developed to automate the specification of K. None, however, have been generally accepted, and most make additional assumptions concerning the distributions being modeled [Zhuang, Huang et al. 1996] or introduce new parameters [McLachlan and Basford 1988]. For generalized projective Gaussian distributions the automated specification of K can be a particularly difficult task. There is in actuality an infinite number of components, and the task becomes one of finding a finite number of components which well approximate that continuum.

Non-optimal global maxima generally occur on the fringes of a model's parameter space. They occur when a component becomes dedicated to a single sample. The variance of this component will tend towards zero while its maximum likelihood value will rapidly increase. Given a likelihood measure, the values at those maxima are actually unbounded. Bayesian methods have been applied in an effort to limit the effects of such maxima [West 1993].

The existence of local maxima is revealed by starting from a different point in parameter space. Since K-means is being used to initialize the parameters, simply changing the order in which the data are supplied causes EM to begin with different initial values and a drastically different arrangement of components results. Figure 2.78 shows the results from FGMM via MLEM using K=7 when the same data used to generate Figure 2.76 is presented in a different order. The effect of the change in the parameters of the components is obvious. A different local maximum has been reached. The TPRs and FPRs for these two different models resulting from different maxima are given in Table 2.19.



Different initialization produces different model versus Figure 2.76
Figure 2.78

	TPR	FPR
Fig. 2.76	0.9222	0.2844
Fig. 2.78	0.9281	0.2829

Different initialization produces different TPR and FPR rates
Table 2.19

The poor consistency resulting from local maxima is revealed by the graphs in Figure 2.79 and 2.80. They also illustrate the importance of the specification of the number of components in determining the accuracy as well as the consistency of the models.

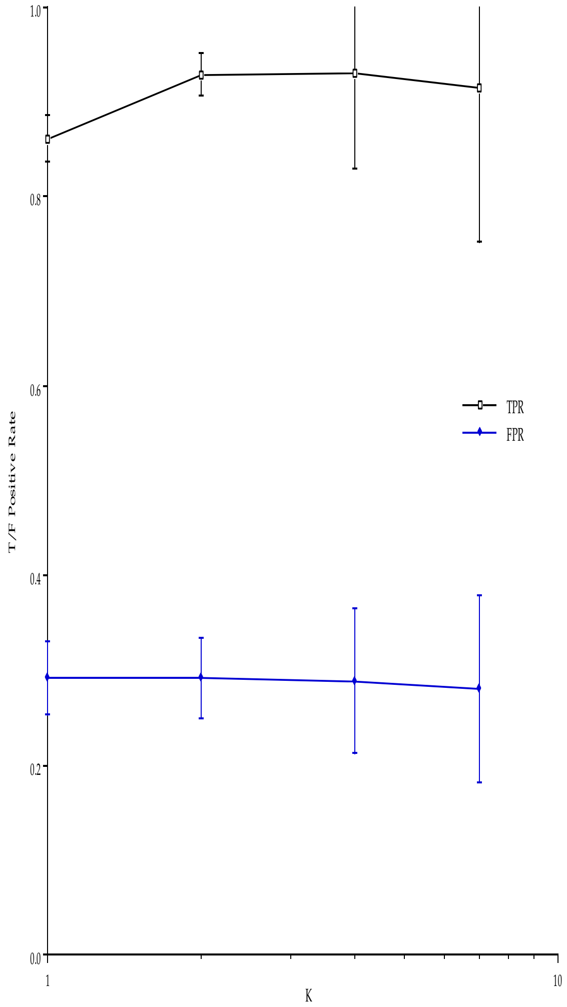
Most researchers recommend developing several models of a set of data from different initial points in parameter space and using a range of K values and then choosing the "best" so as to avoid these problems. In practice such an approach can be time consuming, difficult to implement, and require significant data. The approach to CGMM presented in this dissertation addresses these issues.

Method.	K	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
MM	1	0.86124	0.29245	0.02382	0.03819	0.02854
	2	0.92873	0.29276	0.02216	0.04280	0.03077
	4	0.93117	0.28892	0.10046	0.07593	0.08408
	7	0.91521	0.28005	0.16201	0.09829	0.11829

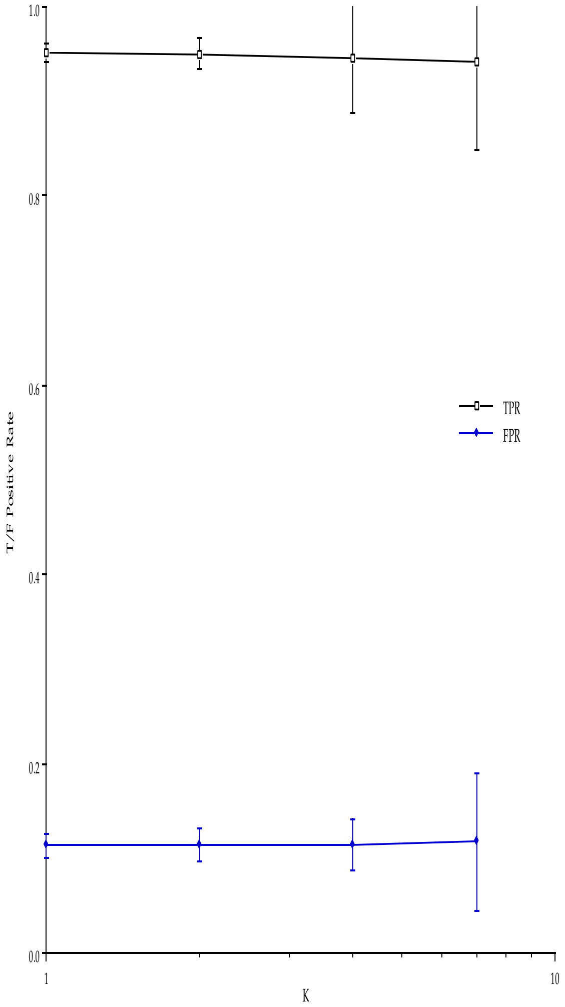
TPRs, FPRs, and consistency for Class B from FGMM classification
Table 2.20

Method.	K	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
MM	1	0.95209	0.11342	0.00972	0.01200	0.01017
	2	0.95060	0.11444	0.01677	0.01742	0.01384
	4	0.94488	0.11365	0.05717	0.02690	0.03465
	7	0.94154	0.11764	0.09238	0.07252	0.07975

TPRs, FPRs, and consistency for Class B' from FGMM classification
Table 2.21



TPR/FPR/Consistency versus N_c for Class B
Figure 2.79



TPR/FPR/Consistency versus N_c for Class B'
Figure 2.80

2.3.7.5. Ease of Analysis

Despite the difficulties with FGMMs, they have received considerable attention and found their way into numerous practical applications [Aylward and Coggins 1994; Bellegarda, Bellegarda et al. 1994; Gish and Schmidt 1994; Samadani 1995; Waterhouse and Robinson 1995]. The most significant advance in terms of quantitative analysis has come from Louis [Louis 1982]. His method for simultaneously defining Fisher's information matrix during the development of the model has enabled the application of a wide range of quantitative analytic techniques. Qualitative analysis is facilitated by the use of Gaussian components. Both the qualitative and quantitative analysis can be especially revealing if its is assumed that the population is actually a mixture of Gaussians. Such mixture models are referred to as direct mixture models. For extruded Gaussian distributions, since the number of components is actually infinite, only indirect FGMMs can be formed.

2.3.8. Summary

Figures 2.81 and 2.82 provide summaries of the performance of the various classifiers analyzed. These figures are plots of TPR versus FPR. Each classifier's average TPR/FPR value is indicated by its abbreviated name. About each average is a circle whose area is proportional to the log of that classifier's combined one-sigma TPR/FPR variance. The larger the circle, the less consistent was that classification technique's accuracy throughout the Monte Carlo simulation.

Four important consistency checks are upheld within these graphs:

1) FGMM with K=1 performs nearly the same as Gaussian classification.

2) KNN, Parzen windows, and MLPs are ordered in accuracy and consistency as their parameters are changed.
2a) The Monte Carlo runs were sufficient in number to capture this expected relation.
2b) The consistency of Parzen windows and KNN improves with increasing neighborhood size/σ / K.
2c) Parzen windowing asymptotically approaches K=1 nearest neighbor as σ approaches 0.

3) Gaussian classifiers provide optimal accuracy and consistency for Problem 2.

4) All techniques provided better accuracy and consistency for Problem 2 versus Problem 1.

These graphs support the following hypothesis:

1) K means and FGMMs via MLEM have poor consistency.

2) For the given problems, K means and FGMMs via MLEM consistencies degrade as the number of components is increased.

2a) The parameter selection process makes poor use of additional resources.

2b) The additional degrees of freedom serve only to confound the problem.

2c) Performance is tied to the user specification of an appropriate number of components, K.

3) For Problem 1, the Monte Carlo runs were insufficient in number to capture the expected order of progression in accuracy as the number of components is varied for K means and FGMMs via MLEM due to the inconsistency of these methods.

Therefore, while FGMMs should provide high levels of accuracy, their development using MLEM results in high levels of inconsistency and reliance on the user specification of K. The graphs clearly show that FGMMs via MLEM provide the most inconsistent levels of accuracy among the classifiers analyzed. An alternate method for GMM development needs to be identified. This dissertation presents such a method.

Method.	K	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
Gauss		0.86057	0.29396	0.01756	0.01737	0.01636
Linear		0.86497	0.31235	0.06153	0.06942	0.03564
KNN	1	0.76094	0.24433	0.02126	0.01861	0.01959
	3	0.82346	0.23834	0.02152	0.01895	0.01949
	7	0.86584	0.24230	0.02137	0.02049	0.01924
	11	0.88107	0.24680	0.02085	0.02146	0.01906
PW	2	0.84960	0.24390	0.02069	0.01960	0.01911
	8	0.90534	0.25473	0.01700	0.01962	0.01650
	16	0.91394	0.28018	0.01458	0.01801	0.01515
	32	0.89889	0.30073	0.01420	0.01750	0.01504
MLP	36	0.90327	0.26495	0.05007	0.05858	0.03610
	108	0.88089	0.24267	0.06650	0.05939	0.04470
	360	0.88044	0.24143	0.06509	0.06155	0.04021
KM	1	0.82714	0.27073	0.01705	0.01739	0.01615
	2	0.86754	0.28483	0.05093	0.03924	0.03897
	4	0.83296	0.24890	0.06654	0.07657	0.07088
	7	0.79113	0.29009	0.05728	0.12592	0.08039
FGMM	1	0.86124	0.29245	0.02382	0.03819	0.02854
	2	0.92873	0.29276	0.02216	0.04280	0.03077
	4	0.93117	0.28892	0.10046	0.07593	0.08408
	7	0.91521	0.28005	0.16201	0.09829	0.11829

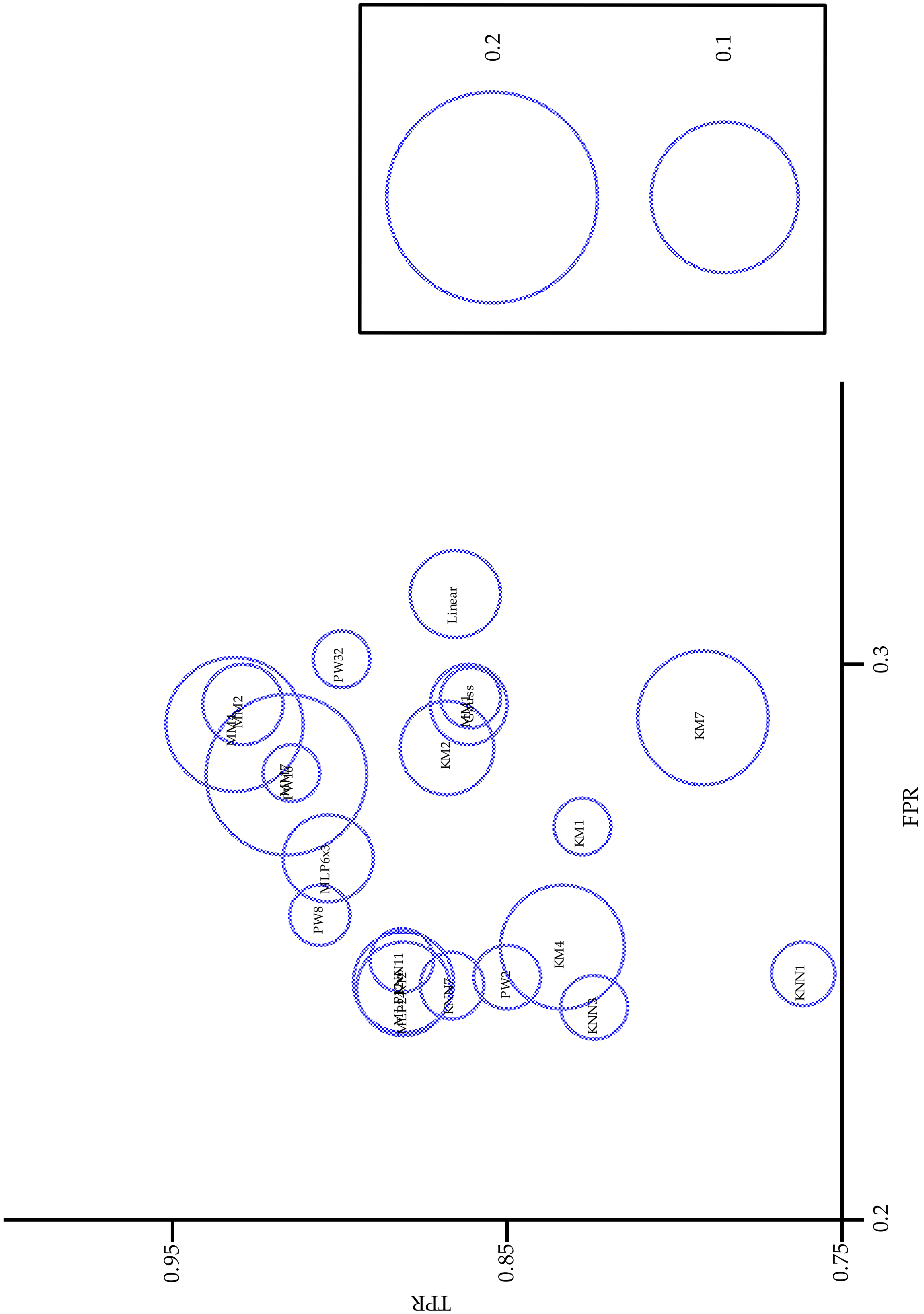
Summary of the recorded Problem 1 TPR, FPR, and one-sigma ranges

Table 2.22

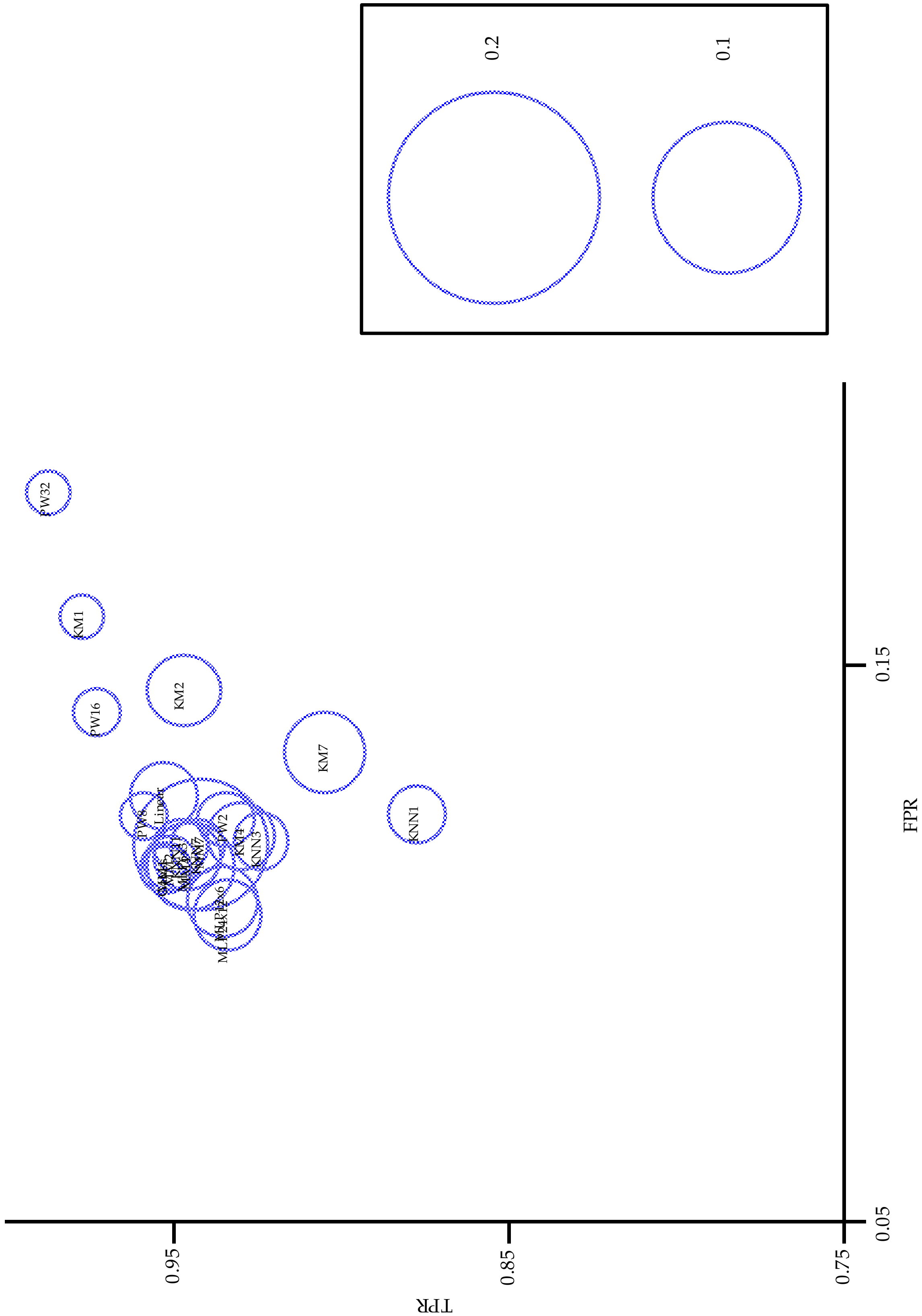
Method.	K	TPR	FPR	TPR One-Sigma	FPR One-Sigma	Comb. One-Sigma
Gauss		0.95211	0.11352	0.00966	0.01185	0.01013
Linear		0.95270	0.12630	0.02965	0.03227	0.02061
KNN	1	0.87729	0.12305	0.01832	0.01420	0.01583
	3	0.92383	0.11823	0.01567	0.01395	0.01404
	7	0.94232	0.11722	0.01426	0.01441	0.01310
	11	0.94764	0.11751	0.01397	0.01481	0.01291
PW	2	0.93420	0.12209	0.01428	0.01476	0.01377
	8	0.95858	0.12271	0.01032	0.01375	0.01089
	16	0.97261	0.14125	0.00728	0.01416	0.00953
	32	0.98743	0.18084	0.00448	0.01625	0.00816
MLP	36	0.94569	0.11549	0.03472	0.03211	0.02013
	108	0.93519	0.10756	0.04273	0.03246	0.02343
	360	0.93376	0.10504	0.03667	0.02764	0.02150
KM	1	0.97706	0.15863	0.00650	0.01508	0.00930
	2	0.94669	0.14549	0.04098	0.01911	0.02298
	4	0.92937	0.11924	0.02515	0.01614	0.01958
	7	0.90416	0.13444	0.03270	0.02753	0.02937
FGMM	1	0.95209	0.11342	0.00972	0.01200	0.01017
	2	0.95060	0.11444	0.01677	0.01742	0.01384
	4	0.94488	0.11365	0.05717	0.02690	0.03465
	7	0.94154	0.11764	0.09238	0.07252	0.07975

Summary of the recorded Problem 2 TPR, FPR, and one-sigma ranges

Table 2.23



Average and one-sigma range of TPR versus FPR for Class B
Figure 2.81



Average and one-sigma range of TPR versus FPR for Class B′
Figure 2.82

2.4. What's Next?

This Chapter has, in fact, presented and analyzed a range of approaches to and implementations of Gaussian mixture modeling. Linear classifiers imply GMMs having just one Gaussian component with a fixed covariance matrix. Gaussian classifiers use GMMs having just one component whose mean and covariance is fit to the training samples of a class. K-means allows multiple Gaussians to represent a distribution, but the covariance matrices of those Gaussian components are fixed. Finally, finite Gaussian mixture modeling uses multiple means to represent a distribution and defines their covariance matrices based on the set of training samples local to that mean. These approaches, however, are limited to using predetermined, finite number of Gaussians, and the parameters of each Gaussian component are determined only in consideration of a global performance measure. The premise of this dissertation is that by using continua of Gaussians parameterized by mean and covariance and constraining the parameterization of those Gaussian components to vary smoothly, the data can be better represented. Such parameterizations exist for the continua of centers and widths for the description of objects in images; they are called medialness cores. The next chapter provides an overview of medialness cores. The subsequent chapter presents the steps necessary to adapt core techniques to the generation of representations of distributions.

2.5. Bibliography

Aylward, S. R. and J. M. Coggins (1994). Spatially Invariant Classification of Tissues in MR Images. Visualization in Biomedical Computing, Rochester, MN,

Babich, G. A. and O. I. Camps (1996). "Weighted Parzen Windows for Pattern Classification." IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(5): 567-574.

Bebis, G. and M. Georgiopoulos (1994). "Feed-Forward Neural Networks." IEEE Potentials **October/November**

Bellegarda, E., J. Bellegarda, et al. (1994). "A Fast Statistical Mixture Algorithm for On-Line Handwriting Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(12): 1227.

Chen, T. and H. Chen (1995). "Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks." IEEE Transactions on Neural Networks **6**(4): 904-910.

Cheng, Y. (1995). "Mean Shift, Mode Seeking, and Clustering." IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(8): 790-799.

Coggins, J. M. (1990). A multiscale description of image structure for segmentation of biomedical images. Visualization in Biomedical Computing, Atlanta, GA, IEEE.

Coggins, J. M. (1992). A Statistical Approach to Multiscale Medial Vision. Mathematical Methods in Medical Imaging, SPIE.

Coggins, J. M. and E. Graves (1994). Geometric Image Analysis Using Multiscale Statistical Features. AAAI Spring Symposium Series: Application of Computer Vision in Medical Image Processing, Stanford University, AAAI Press.

Cohen, A. C. (1967). “Estimation in mixtures of two normal distributions.” Technometrics **9**: 15-28.

Dempster, A., N. Laird, et al. (1977). “Maximum Likelihood for Incomplete Data via the EM Algorithm.” Royal Statistical Society **1**(1):

Duda, R. and P. Hart (1973). Pattern Classification and Scene Analysis. New York, John Wiley and Sons.

Ferguson, T. S. (1983). Bayesian density estimation via mixtures of normal distributions. Recent Advances in Statistics . New York, Academic Press. 287-302.

Fryer, J. G. and C. A. Robertson (1972). “A comparison of some methods for estimating mixed normal distributions.” Biometrika **59**: 639-648.

Gish, H. and M. Schmidt (1994). “Text-Independent Speaker Identification.” IEEE Signal Processing Magazine **11**(4): 18-32.

Hornik, K., M. Stinchcombe, et al. (1989). “Multilayer Feedforward Networks are Universal Approximators.” Neural Networks **2**: 359-366.

Jain, A. and R. Dudes (1988). Algorithms for Clustering Data. Englewood Cliffs, NJ, Prentice Hall.

Jain, A. K. (1989). Fundamentals of Digital Image Processing. Englewood Cliff, NJ, Prentice Hall.

Johnston, B., M. S. Atkins, et al. (1996). “Segmentation of Multiple Sclerosis Lesions in Intensity Corrected Multispectral MRI.” IEEE Transactions on Medical Imaging **15**(2): 154-169.

Jones, M. C., J. S. Marron, et al. (1994). “A brief servey of bandwidth selection for density estimation.” Journal of the American Statistical Association **91**: 401-407.

Jordan, M. I. and L. Xu (1993). Convergence Results for the EM Approach to Mixtures of Experts Architectures. Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Lecocq, C. B. and J.-G. Postaire (1991). Iterations of Morphological Transformations For Cluster Separation Pattern Classification. Symbolic-Numeric Data Analysis and Learning, Versailles, France, Nova Science Publishers, Inc.

Lippmann, R. P. (1987). “An Introduction to Computing with Neural Nets.” IEEE ASSP **3**(4): 4-22.

Loader, C. R. (1995). Local Likelihood Density Estimation. AT&T Bell Laboratories.

Louis, T., A. (1982). “Finding the Observed Information Matrix when Using the EM Algorithm.” Journal of the Royal Statistical Society **44**(2): 226-233.

Mao, J. and A. K. Jain (1995). "Artificial Neural Networks for Feature Extraction and Multivariate Data Projection." IEEE Transactions on Neural Networks **6**(2): 296-317.

McLachlan, G. J. and K. E. Basford (1988). Mixture Models. New York, Marcel Dekker, Inc.

Neter, J., W. Wasserman, et al. (1978). Applied Statistics. Boston, Allyn and Bacon, Inc.

Newcomb, S. (1886). "A generalized theory of the combination of observations so as to obtain the best result." American Journal of Mathematics **8**: 343-366.

Osman, H. and M. Fahmy (1994). "On the Discriminatory Power of Adaptive Feed-Forward Layered Networks." IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(8): 837.

Parzen, E. (1962). Stochastic Processes. San Francisco, Holden-Day.

Pearson, K. (1894). "Contribution to the mathematical theory of evolution." Phil. Trans. Roy. Soc. A **185**: 71-110.

Peterson, G. F., D. C. St. Clair, et al. (1995). "Using Taguchi's Method of Experimental Design to Control Errors in Layered Perceptrons." IEEE Transactions on Neural Networks **6**(4): 949-961.

Poggio, T. and F. Girosi (1990). "Networks for Approximation and Learning." Proceedings of the IEEE **78**(9): 1481-1497.

Press, W. H., B. P. Flannery, et al. (1990). Numerical Recipes in C. Cambridge, Cambridge University Press.

Samadani, R. (1995). "A Finite Mixtures Algorithm for Finding Proportions in SAR Images." IEEE Transactions on Image Processing **4**(8): 1182-1186.

Schalkoff, R. (1992). Pattern recognition: statistical, structural and neural approaches. New York, John Wiley & Sons, Inc.

Silverman, B. W. (1978). "Choosing the window width when estimating a density." Biometrika **65**(1): 1-11.

Silverman, B. W. (1986). Density Estimation for Statistics and Data Analysis. London, Chapman and Hall.

Sobol', L. M. (1994). A Primer for the Monte Carlo Method. Boca Raton, CRC Press.

Speckman, P. (1988). "Kernel Smoothing in Partial Linear Models." Journal of the Royal Statistical Society **50**(3): 413-436.

Tan, W. Y. and W. C. Chang (1972). "Some comparisons of the method of moments and the method of maximum likelihood in estimating parameters of a mixture of two normal densities." Journal of the American Statistical Association **67**: 702-708.

Touzani, A. and J. G. Postaire (1988). "Mode Detection by Relaxation." IEEE Transactions on Pattern Analysis and Machine Intelligence **10**(6): 970-978.

Waterhouse, S. R. and A. J. Robinson (1995). Non-linear Prediction of Acoustic Vectors Using Hierarchical Mixtures of Experts. Neural Information Processing Systems, 7, MIT Press Cambridge MA.

Wells III, W. M., W. E. L. Grimson, et al. (1996). "Adaptive Segmentation of MRI Data." IEEE Transactions on Medical Imaging **15**(4): 429-442.

West, M. (1993). "Approximating Posterior Distributions by Mixtures." Journal of the Royal Statistical Society **55**(2): 409-422.

Xu, L. and M. I. Jordan (1995). On Convergence Properties of the EM Algorithm for Gaussian Mixtures. Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Xu, L., A. Krzyzak, et al. (1994). "On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates, and Receptive Field Size." Neural Networks **7**(4): 609-628.

Zhuang, X., Y. Huang, et al. (1996). "Gaussian Mixture Density Modeling, Decomposition, and Applications." IEEE Transactions on Image Processing **5**(9): 1293-1302.

Chapter 3

MEDIALNESS CORES

Like the core of an apple.

- Pizer, 1995

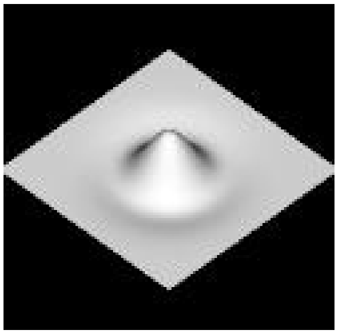
Medialness cores capture the location, size, and shape of objects in images. They are the loci of generalized local maxima in a function called medialness defined on the scale space of an image[1]. To understand cores requires understanding medialness kernels, medialness space, and generalized local maxima. This chapter provides a brief overview of these concepts and the process of medialness core extraction.

The following discussion borrows heavily from two papers on medialness cores. The first paper [Pizer, Eberly et al. 1996] provides a detailed description of the mathematics and the invariance properties of medialness cores. The second paper [Morse, Pizer et al. 1996] describes the insensitivity of medialness cores to image disturbances such as noise and blurring.

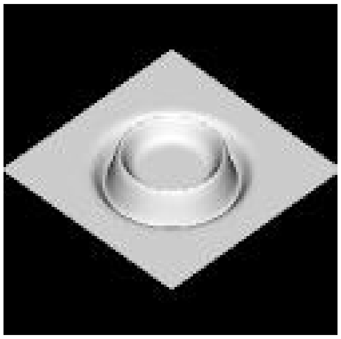
3.1. Medialness Kernels

Medialness can be measured at a spatial image location $\underline{x}$ and scale σ via application of a medialness kernel. Medialness kernels operate by integrating boundariness measurements at a radial distance proportional to scale, so medialness kernels are multilocal. They yield responses that are particularly strong along the track of an object's local spatial centers at scales proportional to the object's local widths. Medialness kernels can be distinguished by the focus of their medialness measurements and by their dynamics.

[1] Hereafter the term medialness space will be used to refer to the domain, the set of all image locations $\underline{x}$ and scales σ, on which a medialness function is defined.



Laplacian of a Gaussian;
a central medialness kernel
Figure 3.1



Morse medialness;
an offset medialness kernel
Figure 3.2

A medialness kernel's focus can be central or offset. Medialness kernels having a central focus are band pass spatial filters centered about their query point $\underline{x}$. Medialness kernels having an offset focus only consider data a fixed distance from $\underline{x}$.

Two fixed and oriented medialness kernels in 2D are depicted in Figures 3.1 and 3.2.

The dynamics of a medialness kernel can be characterized as fixed, oriented, or adaptive. Medialness kernels with fixed dynamics maintain a radially symmetric shape throughout the core extraction process. They can be applied to an entire image using convolution. Oriented medialness kernels also have a fixed shape, but, they are not radially symmetric and the orientation at which they are applied at each point in an image is dependent on the local image data and the local tangent frame of the core. Adaptive medialness kernels' shape and orientation depend on the local data and the local tangent frame of the core.

Fritsch [Fritsch, Pizer et al. 1994; Fritsch, Eberly et al. 1995] has investigated the fixed-central, Laplacian-of-Gaussian (LoG) medialness kernel (Equation 3.1). The equation of the LoG medialness kernel is

$$K(\underline{x},\sigma) = \frac{N\sigma^2 + \|\underline{x}\|^2}{\sigma^2} \mathbf{G}(\underline{x},\sigma) \qquad [3.1]$$

where $\underline{x} \in \Re^N$, $\subset\subset\underline{x}\subset\subset$ denotes the length of $\underline{x}$, and $\mathbf{G}(\underline{x},\sigma)$ is a unit normalized Gaussian (Equation 2.1).

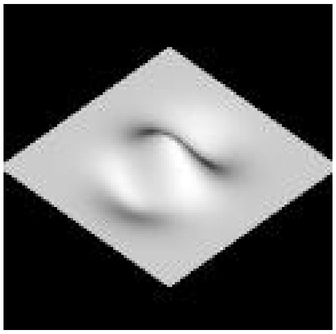
It can be applied at every position in an image using convolution to create the central-fixed medialness function at scale σ. Varying σ allows one to construct the medialness space.

$$F(\underline{x},\sigma) = \mathbf{I}(\underline{x}) * K(\underline{x},\sigma) \qquad [3.2]$$

where * denotes convolution.

An example of an oriented-central medialness kernel for one direction is shown in Figure 3.3 [Fritsch, Pizer et al. 1994; Fritsch, Eberly et al. 1995]. The medialness function obtained from this kernel is defined as

$$F(\underline{x},\sigma) = -\sigma^2 L_{\underline{p}\underline{p}} = -\sigma^2 \underline{p}^t D^2 L \underline{p} = -\sigma^2 \underline{p}^t \underline{\underline{H}} \underline{p} = -\sigma^2 \alpha \qquad [3.3]$$



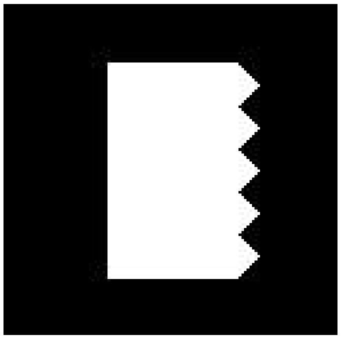
An oriented-central medialness function for a single direction
Figure 3.3

where ^{t} denotes transposition, $\underline{\underline{H}}$ is the Hessian of the image measured at spatial location $\underline{x}$ and scale σ, α is the largest magnitude negative eigenvalue of D^2L, and $\underline{p}$ is the corresponding eigenvector of D^2L. At a single spatial location and scale, the kernel is oriented in the direction $\underline{p}$ which is determined by the image data, i.e., maximum over orientation of the 2nd directional derivative of image intensity at scale σ.

Adaptive-offset medialness kernels are currently undergoing extensive investigation by Matt McAuliffe at the University of North Carolina, Chapel Hill [Fritsch, Eberly et al. 1995]. Adaptive-central medialness kernels are most closely related to the chosen implementation of GGoF kernels. Adaptive-central medialness kernels, however, have not been investigated for the definition of medialness cores.

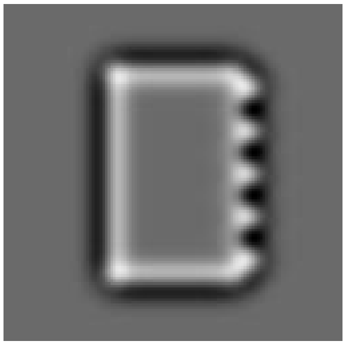
3.2 Medialness Space

A medialness space of an image consists of the values of a medialness function for a range of spatial locations $\underline{x}$ and scales σ. This space is not Euclidean, i.e., $\underline{x}$ and σ are not of commensurate units. Derivative measurements made with respect to scale or at different scales must account for the structure of the space [Eberly 1996; Pizer, Eberly et al. 1996].

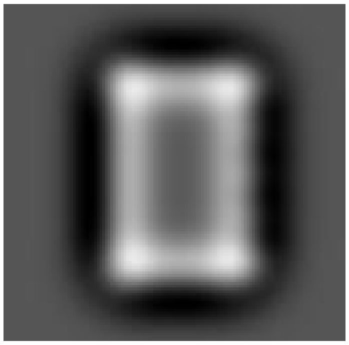


An image of a binary object
Figure 3.4

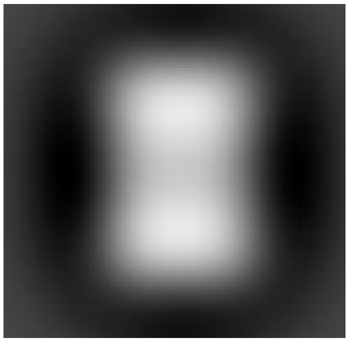
Select slices with respect to scale of the central-linear, LoG medialness space of the binary object shown in Figure 3.4 are given in Figures 3.5 through 3.7. Notice at small scale the middles of the individual sawteeth are well localized as maxima as are the corners of the rectangle. At medium scale the points more interior to the rectangle's corners produce the highest response. The details of the sawteeth are barely visible. At large scale the maxima extending from the corners have nearly merged.



LoG Medialness of Figure 3.4 at small scale
Figure 3.5.

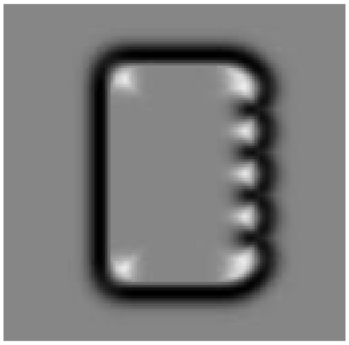


LoG Medialness of 3.4 at medium scale
Figure 3.6

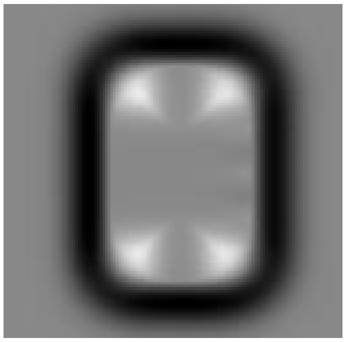


LoG Medialness of 3.4 at large scale
Figure 3.7

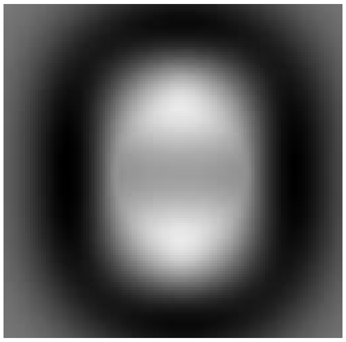
Select slices with respect to scale of a medialness space generated from a central-oriented medialness kernel (Equation 3.3) of the binary object shown in Figure 3.4 are given in Figures 3.8 through 3.10.



Central-oriented medialness of Figure 3.4 at small scale
Figure 3.8



Central-oriented medialness of Figure 3.4 at medium scale
Figure 3.9



Central-oriented medialness of Figure 3.4 at large scale
Figure 3.10

3.3 Medialness Cores: Generalized Maxima and Height Ridges

Generalized maxima are points which are local maxima in a not necessarily proper subset of basis directions. Eberly [Eberly 1996] provides a more detailed discussion of these geometric constructs which are called height ridges. Interesting research into an alternate method for extracting of the ridges of functions is being conducted by Jacob Furst at The University of North Carolina, Chapel Hill.

A point $\underline{y} \leftarrow^{N+1}$ is a generalized local maximum in a set of basis directions *V*, i.e., $\underline{v}^{(i)} \bullet \underline{v}^{(j)}=0$ for all i•j where $V=\{\underline{v}^{(i)} \leftarrow^{N+1}; i=1..N+1-M\}$, of a function **F** when

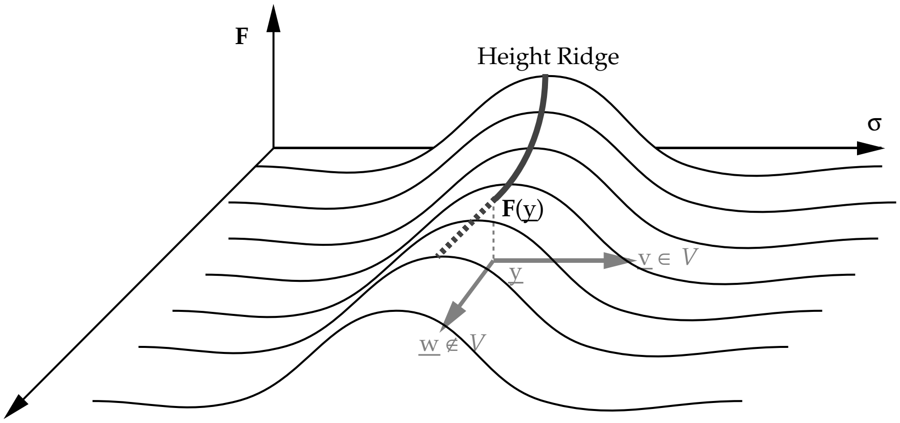
$$\underline{v}^{t}D\mathbf{F}(\underline{y})=0 \quad \text{and} \quad \underline{v}^{t}D^{2}\mathbf{F}(\underline{y})\underline{v}<0 \quad \text{for all } \underline{v} \text{ in } V. \qquad [3.4]$$

where D is the gradient operator and D^2 is the Hessian operator.

When M=0 then such points are strict local maxima. When M>0 then such points are generalized local maxima of dimensionality M.

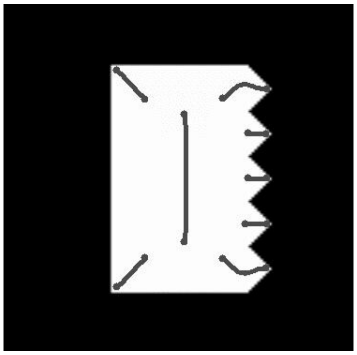
Points on a maximum convexity height ridge are generalized local maxima in the directions of largest convexity. Second derivative information at a point $\underline{y}$ is captured by the Hessian of **F** at that point, $\underline{\underline{\mathbf{H}}} = D^2\mathbf{F}(\underline{y})$. For an M-dimensional height ridge, the N+1-M most negative eigenvalued eigenvectors of the Hessian are the directions of greatest convexity. If all of those eigenvalues are negative, those eigendirections specify the set of directions *V* for the equations of generalized local maxima given above, i.e., $|V| = N+1-M$.

In Figure 3.11, the point $\underline{y}$, and the directions $\underline{v}$, and $\underline{w}$ exist in the plane spanned by x and σ. **F** is a height surface above that plane. The eigenvectors of the Hessian of **F** at $\underline{y}$ are $\underline{v}$ and $\underline{w}$. The direction $\underline{v}$ has the most negative eigenvalue and thus is approximately normal to the ridge, i.e., $V = \{\underline{v}\}$.

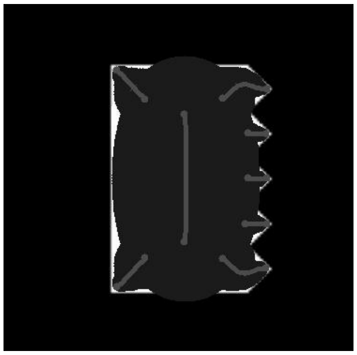


An M=1 dimensional height ridge V={$\underline{v}$} in $\leftarrow^{N+1=2}$
Figure 3.11

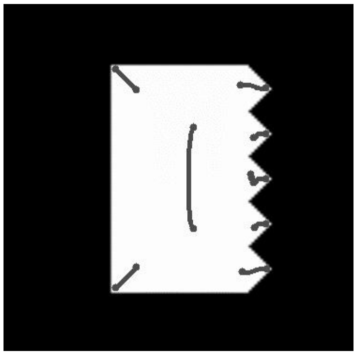
When the function **F** is a medialness function, $\underline{y}=(\underline{x}, \sigma)$ and the height ridges are medialness cores of the corresponding image. Visualizations of some of the 1-dimensional (M=1) medialness cores of Figure 3.4 are shown in Figures 3.12 through 3.15. Figures 3.12 and 3.14 correspond to the spatial projections of the LoG and $L_{\underline{pp}}$ medialness cores. Figures 3.13 and 3.15 illustrate the scales associated with the cores via circles of appropriate radius (radius = ρσ where ρ=1.0) centered on the corresponding spatial projection locations.



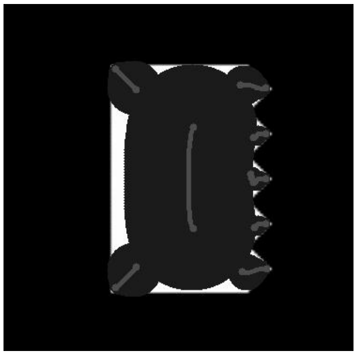
Spatial projection of LoG Medialness cores of Figure 3.4
Figure 3.12



Filled circles defined by scale component of LoG Medialness cores of Figure 3.4
Figure 3.13



Spatial projection of a central-oriented Medialness cores of Figure 3.4
Figure 3.14



Filled circles defined by scale component of a central-oriented Medialness cores of Figure 3.4
Figure 3.15

The LoG medialness function has proven to be useful for defining cores of anatomic objects with nonparallel sides, approximately uniform interiors, edges of fixed contrast polarity, and possibly low signal to noise ratio. The $L_{\underline{pp}}$ medialness function has proven to be useful for defining objects with parallel sides, uniform interior intensity, and possibly intensity variations along the central skeleton. The same process was used to extract all of the cores shown. It is detailed in the following section.

3.4. Height Ridge Extraction

The extraction of M-dimensional cores of objects in N-dimensional images using a medialness function **F** proceeds in two phases: flowing from a starting "stimulation" point to an (N+1-M)-dimensional height ridge point and then traversing that height ridge. The domain of medialness cores is non-Euclidean, i.e., $\leftarrow^{N}x\leftarrow^{+}$, and thus requires the adjustment of measures made at different scales. Here we present the general case in which the domain of the height ridge is Euclidean, i.e., $\leftarrow^{N+1}$. This is the case which is employed by GGoF cores.

Define

$\alpha^{(i)}$ the ascending ordered eigenvalues of $D^2\mathbf{F}(\underline{y})$ $\quad i = 1..N+1$

$\underline{v}^{(i)}$ the correspondingly ordered eigenvectors of $D^2\mathbf{F}(\underline{y})$ $\quad i = 1..N+1$

and the directional derivatives

$$\mathbf{P}^{(i)}(\underline{y}) = \underline{v}^{(i)^t} D\mathbf{F}(\underline{y}) \qquad i = 1 .. N+1 \qquad [3.5]$$

Then the following conditions must hold at $\underline{y}$ for a maximum convexity height ridge to exist at that point on the surface **F**

$$\alpha^{(N+1-M)} < 0 \qquad \text{and} \qquad \mathbf{P}^{(i)}(\underline{y}) \cong 0 \quad \text{for all } i=1..(N+1-M) \qquad [3.6]$$

The conditions of equality to zero are conveniently tested by formulating the function

$$\mathbf{J}(\underline{y}) = \sum_{i=1}^{N+1-M} \left(\mathbf{P}^{(i)}(\underline{y})\right)^2 \qquad [3.7]$$

and testing $\mathbf{J}(\underline{y})$<tolerance, e.g., tolerance = 10^{-4}.

3.4.1. Flowing to a Height Ridge

Given an initial (user-specified) starting point, $\underline{y}^{(0)}$, its associated ridge can be found using a conjugate directions search with respect to the $D^2\mathbf{F}(\underline{y}^{(0)})$ so as to minimize $\mathbf{J}(\underline{y})$. That is, a line search is performed from $\underline{y}^{(0)}$ in the direction $\underline{v}^{(0)}$, and if the minimum in that direction is not sufficient, from that point the direction $\underline{v}^{(1)}$ is searched. If after N+1 iterations the resulting minimum of $\mathbf{J}(\underline{y})$ is not within tolerance or if $\alpha^{(N+1-M)}$ at $\underline{y}$ is not less than zero, a new stimulation point is required. For the examples in this dissertation, Brent's method is used to perform the line searches [Press, Flannery et al. 1990].

3.4.2. Traversing the Height Ridge

Once a ridge point is found, instead of explicitly calculating the tangents of the ridge, a step can be taken in the approximate tangent frame directions and then a flow to the ridge is performed if that point is too far off the actual ridge, i.e., $\mathbf{J}(\underline{y})$ is larger than tolerance. [Eberly 1996]

The tangent frame is well approximated by the remaining, i.e., the M largest eigenvalued, eigenvectors of $D^2\mathbf{F}(\underline{y})$. By stepping in these approximately tangent directions and using the flow algorithm if $\mathbf{J}$ becomes large, the height ridge can be traversed using only the eigenvalues and eigenvectors of the Hessian. This technique circumvents many of the difficulties associated with the discontinuities common in the eigenvector fields of functions. In fact, it is sufficient to test and correct for the swapping of the first L-M eigenvectors between consecutive ridge points to handle the remaining discontinuities [Eberly 1996].

For the examples presented, the traversal step size was 0.1 pixel units. Termination of ridge traversal occurs when the ridge criteria are no longer upheld. Ridges are also allowed to turn less than $\pi/8$ compared to the previous step direction. This rule halts the integration of subcores and rarely interferes with normal core traversal.

3.5 Overview of Insensitivities

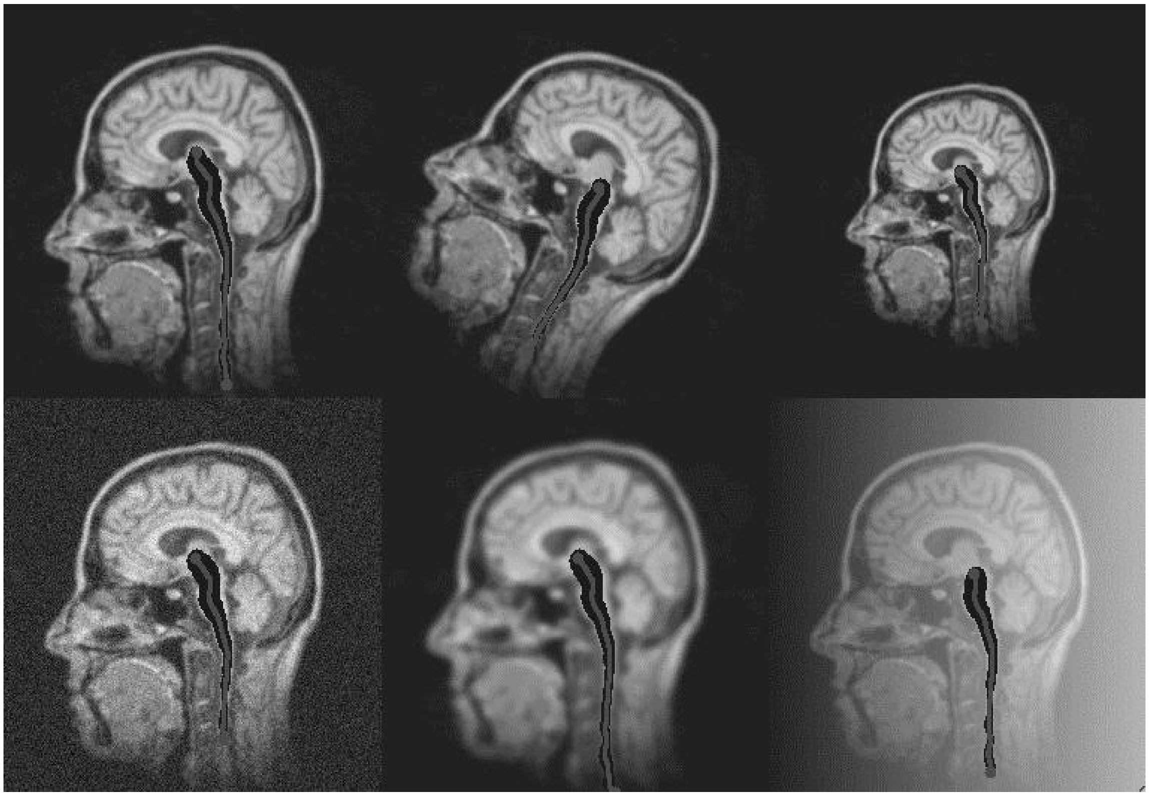
Medialness cores have been proven to be insensitive to a wide variety of object disturbances and image noise due to its use of multilocal boundariness measures such as medialness functions applied at a range of scales, i.e., aperture sizes. Figure 3.16 at the end of this chapter illustrates the consistent extraction of a 1D, central-linear, LoG medialness core of the brain stem from a 2D magnetic resonance image which has undergone various deformations.

3.6. Summary

Medialness cores capture the location, size, and shape of objects via continuous representations of their central tracks and local widths. Such representations are formed using multilocal, medialness kernels. A variety of medialness kernels exist, each with its own strengths and weaknesses. Medialness cores have been proven to be invariant to a variety of object disturbances, e.g., rotation, translation, and scale, and insensitive to a variety of image noise, e.g., changes in absolute intensity.

This dissertation will exploit medialness core definition and extraction techniques for the generation of representations of extruded Gaussian distributions in feature space. Specifically, Gaussian goodness-of-fit cores will represent extruded Gaussian distributions by tracking the continuum of means of those distributions and estimating the local variance of the distribution normal to that track.

Many of the invariances and insensitivities of medialness cores are beneficial in representing the distribution of samples in feature space. Rotation and translation invariance are two such qualities. In feature space, however, a medialness function needs to be sensitive to changes in intensity rather than to boundariness. For example, skewness and other moments of sample frequency in feature space must be modeled. GGoF functions consider such information as well as the other information necessary for density estimation. The next Chapter evaluates the accuracy and consistency of the maxima of GGoF functions.



Medialness core of brainstem is extracted
invariant to rotation, scale, noise, blur, and intensity variations
Core spatial projections shown in dark gray.
Core scale indicated by overlaid filled circle of radius $\rho\sigma$ (ρ=1.0).
Figure 3.16

3.7. Bibliography

Eberly, D. (1996). Ridges in Image and Data Analysis. Dordrecht, Kluwer Academic Publishers.

Fritsch, D. S., D. Eberly, et al. (1995). Stimulated Cores and their Applications in Medical Imaging. IPMI 1995: Information Processing in Medical Imaging, Kluwer Series in Computational Imaging and Vision.

Fritsch, D. S., S. M. Pizer, et al. (1994). Cores for Image Registration. Medical Imaging '94: Image Processing, SPIE.

Morse, B. S., S. M. Pizer, et al. (1996). "Zoom-Invariant Vision of Figural Shape: Effects on Cores of Image Disturbances." Computer Vision and Image Understanding *Submitted*

Pizer, S. M., D. Eberly, et al. (1996). "Zoom-invariant Vision of Figural Shape: the Mathematics of Cores." Computer Vision and Image Understanding *Submitted*

Press, W. H., B. P. Flannery, et al. (1990). Numerical Recipes in C. Cambridge, Cambridge University Press.

Chapter 4

GOODNESS-OF-FIT FUNCTIONS

Goodness of fit is concerned with assessing the validity of models involving statistical distributions.

- Rayner, 1989

This chapter is concerned with the consistency and accuracy of three popular univariate Gaussian goodness-of-fit functions. Consistency and accuracy is quantified by evaluating the correspondence between the μ and σ values producing a local maximum in Gaussian goodness-of-fit and the actual μ and σ of the sampled population. Monte Carlo simulations are used to reveal the effects of the number of samples, population skewness, and binning technique on such maxima. That analysis leads to the selection of the loglikelihood function using overlapped-equiprobable binning for the generation of accurate and consistent Gaussian goodness-of-fit cores.

4.1. Goodness-of-Fit Measures

The validity of a distribution model is assessed using a null hypothesis test. The null hypothesis for a goodness-of-fit (GoF) function is that there is no difference between two distributions except chance differences due to finite sampling. Methods for testing this hypothesis include 1) omnibus procedures 2) likelihood ratio tests involving specific alternatives, 3) measures of moments: skew, and/or kurtosis, and 4) graphical procedures. With the goal of automatically developing CGMMs of unknown distributions using these tests, omnibus procedures using an expected Gaussian distribution are most appropriate. These tuned functions will be referred to as Gaussian goodness-of-fit (GGoF) functions. Omnibus GoF procedures can be grouped as those based on χ^2 measurements and those based on empirical distribution functions (EDF). [Koziol 1986]

χ^2 Measures: Pearson's idea when developing his (the original) χ^2 measure was to reduce the problem of GoF to the simpler problem of comparing observed bin frequencies, $O^{(i)}$, with expected bin frequencies, $E^{(i)}$. Three popular examples of χ^2 GoF functions are Pearson's χ^2 (Equation 4.1), Read and Cressie's power divergent statistic (Equation 4.2), and the log likelihood ratio (Equation 4.3). [Read and Cressie 1988]

$$\chi_P^2 = \sum_{i=1}^{B} \frac{\left(O^{(i)} - E^{(i)}\right)^2}{E^{(i)}} \qquad [4.1]$$

$$\chi_{R\&C}^2 = \frac{9}{5} \sum_{i=1}^{B} O^{(i)} \left(\left(\frac{O^{(i)}}{E^{(i)}} \right)^{2/3} - 1 \right) \qquad [4.2]$$

$$\chi_{LLR}^2 = 2 \sum_{i=1}^{B} O^{(i)} \,\mathbf{ln}\left(\frac{O^{(i)}}{E^{(i)}} \right) \qquad [4.3]$$

EDF Measures: EDF measures are based on the fact that "if one plots the ordered univariate sample versus the corresponding percentiles of the standard normal distribution, one should observe approximately a straight line if the sample indeed is normally distributed." [Koziol 1986] Examples of EDF statistics include Cramer-von Mises based statistics such as the Cramer-von Mises statistic (Equation 4.4) and the Anderson-Darling statistic (Equation 4.5), the Shapiro-Wilk statistics, and the Kolmogorov-Smirnov statistics, e.g., Equations 4.6-4.8. These measures use the ordered values $z^{(i)}$, the cumulative Gaussian distribution's value at the ordered sample values. [Stephens 1974]

$$W^2 = \sum_{i=1}^{|S|} \left(z^{(i)} - \frac{2i-1}{2|S|} \right)^2 + \frac{1}{12|S|} \qquad [4.4]$$

$$A^2 = -\frac{1}{|S|} \sum_{i=1}^{|S|} (2i-1)\left(\mathbf{ln}\left(z^{(i)}\right) + \mathbf{ln}\left(1 - z^{(|S|+1-i)}\right) \right) - |S| \qquad [4.5]$$

$$D^+ = \max_{1 \le i \le |S|} \left(\frac{i}{|S|} - z^{(i)} \right) \qquad [4.6]$$

$$D^- = \max_{1 \le i \le |S|} \left(z^{(i)} - \frac{(i-1)}{|S|} \right) \qquad [4.7]$$

$$D = \max_{1 \le i \le |S|} \left(D^{+}, D^{-} \right) \tag{4.8}$$

Compared to χ^2 statistics, EDF statistics are generally more computationally expensive, require a problematic ordering of multivariate samples, and have not been as well evaluated on discrete data. Since this dissertation is concerned with discrete data and since the GGoF functions are applied repeatedly to extract a single GGoF core, further GGoF evaluation is limited to χ^2 methods.

In this dissertation, GGoF functions are applied to a set of data for a range of parameters m and s values. The set of parameters producing the maximum GGoF value will ideally correspond with the parameters of the sampled data's underlying population. Chi-squared functions are explained in the following section. Subsequent sections detail the consistency and the accuracy of the correspondence between the parameters of their maxima and a sampled Gaussian's actual parameters.

4.2. χ^2 Gaussian Goodness-of-Fit

χ^2 functions test the null hypothesis that the samples $x^{(i)}$, i=1..$|S|$, are well represented by the density function $\mathbf{F}(x)$. Partitioning the random samples $x^{(i)}$ into B cells of ranges $R^{(j)}$, j=1..B, produces observed frequencies $O^{(j)}$. The null hypothesis is true when the $O^{(j)}$ have a binomial distribution with parameters $O = \left\{ O^{(k)} \right\}_{k=1}^{B}$ and

$$E^{(j)} = P\left(x^{(i)} \text{ falls in } R^{(j)} \right) = \int_{R^{(j)}} d\,\mathbf{F}(x) \tag{4.9}$$

Thus, the GoF problem reduces to one of testing whether a multinomial distribution, O, has cell probabilities $E^{(j)}$. Pearson showed that the quantities $O^{(i)}$–$E^{(i)}$ have an approximately multivariate normal distribution and that the quadratic of Equation 4.9 is approximately χ^2 distributed with B-1 degrees of freedom, χ^2_{B-1}. That is,

$$P\left(\chi^2_P \ge c \right) \rightarrow P\left(\chi^2_{B-1} \ge c \right) \qquad \text{for any } c \bullet 0 \tag{4.10}$$

and thus

$$P\left(\chi_P^2 \geq \chi_{B-1}^2(\alpha)\right) \to \alpha \qquad [4.11]$$

This fact is independent of whether **F** is univariate or multivariate, discrete or continuous. Fisher noted that the log likelihood ratio statistic, χ_{LLR}^2, is asymptotically equivalent to Pearson's χ_P^2. Read and Cressie have proven the same for their statistic, $\chi_{R\&C}^2$. Thus, rejection of model validity occurs when the observed value of a χ^2 function is greater than or equal to the χ^2 distribution value found in the $\chi_{B-1}^2(\alpha)$ tables for a pre-specified percentage point, i.e., α*100%.

One important consideration is the "smoothness" of these functions. Smooth GoF functions are "constructed so as to have good power against alternatives whose probability density functions depart smoothly from the desired ... Smooth changes include shifts in mean, variance, skew, and kurtosis." [Rayner and Best 1989] The smoothness criterion is important for the algorithms of this dissertation so that the resulting GGoF space is differentiable and its extrema are well localized. Moment-based GoF functions utilizing Hermite polynomials directly address this smoothness criterion, but the use of such functions is computationally expensive [Rayner and Best 1989]. The three tests being discussed have also been proven to be smooth functions [Cressie and Read 1984]. My experiments have shown that the binning technique also plays a significant role in the smoothness of the GGoF function.

4.3. Univariate Binning

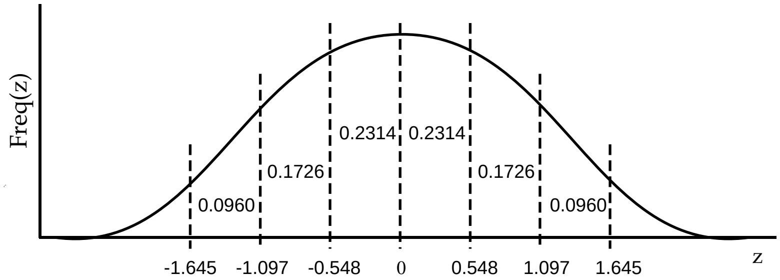
The allocation of the samples to cells is "binning." A binning technique is defined by the number of bins and their feature space ranges.

Number of Bins: Numerous researchers have devised formulae for suggesting the number of bins for which the various χ^2 tests will provide optimal power. It is generally accepted that a test's power will increase as B is increased up to a point, and then for larger B values the power will begin to decrease. Dahiya and Gurland [Dahiya and Gurland 1973] suggest that frequently 4 or 5 bins are appropriate. Most researchers agree that the optimal number is usually quite small but may increase as the number of samples increases [Read and Cressie 1988]. For this dissertation, B=6. However, additional research in this area is warranted given the use of GGoF measures in this dissertation.

Bins can cover equal ranges in feature space, have equal probability, or deviate from equiprobable using a specified weighting [Koziol 1986]. This dissertation also considers an alternate binning strategy based on overlapping bins [Ivchenko and Tsukanov 1984; Hall 1985] which has been shown to increase the accuracy of GoF estimates.

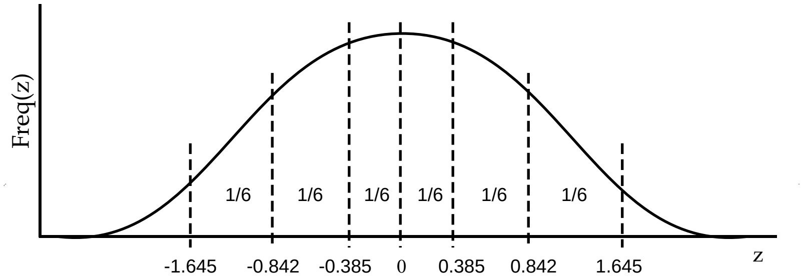
For this dissertation, each binning technique only bins samples within a fixed range of feature space, μ±1.645σ; effectively, the sampled distributions are being compared to clipped Gaussians. This range theoretically captures 90% of the relevant samples while limiting interference from neighboring clusters and speeding computations. Preliminary research indicates that this clipped Gaussian extent is sufficient for the purposes of this dissertation, but additional research should focus on quantifying the effect.

Equirange Bins: Pearson's original work [Pearson 1894] used bins having equal feature space ranges. Such binning is independent of the distribution being considered. Consider the Gaussian distribution shown in Figure 4.1 which is partitioned into 6 equal spatial range bins spanning . It has been suggested by numerous authors that for bins of unequal probabilities, enough samples should be considered so that each bin has an expected frequency greater than two. Given the expected probabilities shown, at least 21 samples are needed. This can be a limiting factor.



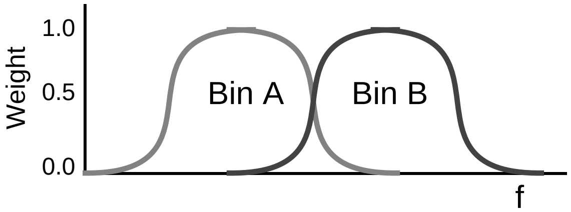
Allocation of the ±1.645σ extent of a Gaussian to 6 equirange bins. Expected bin probabilities are listed inside the bins.
Figure 4.1

Equiprobable Bins: It is generally accepted, following the work by Mann and Wald [Mann and Wald 1942], that equiprobable bins provide the best power for most situations in which the parameters of the expected distribution are known, i.e., not estimated from the samples. The evaluation of a GGoF space is carried out using known parameters. That is, GGoF space defines the μ and σ, not the data. The allocation of a ±1.645σ extent of a Gaussian to 6 equiprobable bins is shown in Figure 4.2. This binning is dependent on the expected distribution. Because of the increase in power it is generally accepted that each bin should have an expected frequency of greater than one, requiring only 7 samples.



Allocation of the 1.645σextent of a Gaussian to 6 equiprobable bins. Expected bin probability is listed inside the bins.
Figure 4.2

Weighted Binning: Weighted binning can be used to tune bin size to the problem at hand. Such adaptive behavior is not studied in this chapter, but Chapter 5 introduces a technique, adaptive-normal GGoF, which can extend any GGoF function / binning technique so that their multivariate extent is based on the local distribution of samples (Section 5.2.3.3).



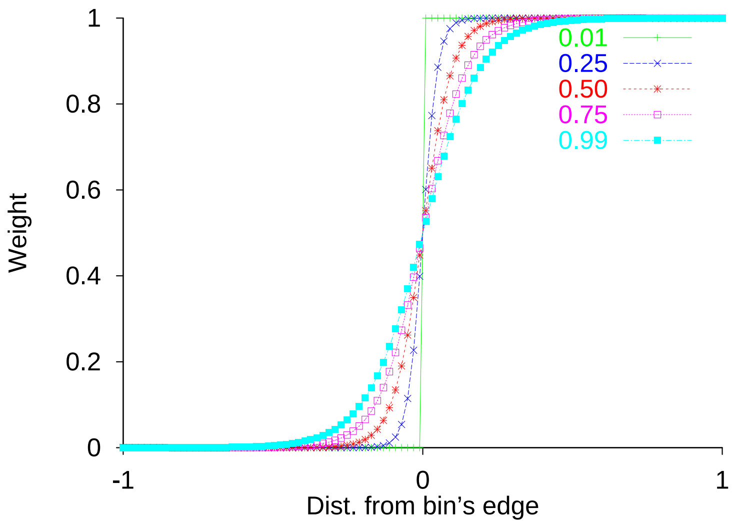
Weighting of samples with respect to two bins using overlapped binning
Figure 4.3

Overlapped-Equirange/Overlapped-Equiprobable Binning: GGoF spaces generated using equirange or equiprobable binning contain "structures" due to sampling which persist through σ. These unwanted structures are caused by the abrupt transition of samples from one bin to the next due to small changes in μ. As a result, a technique for smoothing the transition of the samples between the bins was considered. It operates by weighting each sample with respect to each bin [Ivchenko and Tsukanov 1984; Hall 1985]. For this dissertation a novel weighting function was devised. Figure 4.3 illustrates this weighting scheme for two adjacent bins.

The sample weighting is determined using paired, opposing sigmoidal functions (Equation 4.12) to delineate each bin. The parameter Ω was added so that the amount of overlap between the bins could be controlled. The variable y is the distance of the sample from the bin's edge.

$$W(y,\Omega)=\frac{1}{1+e^{-\left(y/\ln(1+0.1*\Omega)\right)}} \qquad [4.12]$$

This function is plotted in Figure 4.4 for a variety of values for Ω and y.



Sigmoidal weighting function for a variety of Ω values.
Figure 4.4

The distance of a sample from a bin's edge is normalized with respect to half the width of the shortest abutting bin. If a sample x is being weighted with respect to bin j and bin j covers the range $R^{(j)} = f^{(j)}_{Max} - f^{(j)}_{Min}$ then

$$R^{(j)}_{-Min} = MIN(R^{(j)}, R^{(j-1)}) \quad [4.13]$$

$$R^{(j)}_{+Min} = MIN(R^{(j)},R^{(j+1)}) \quad [4.14]$$

and the weight of x with respect to bin j is

$$W^{(j)}(x) = W\left(\frac{x - f^{(j)}_{Min}}{0.5 * R^{(j)}_{-Min}}, \Omega\right) * W\left(\frac{f^{(j)}_{Max} - x}{0.5 * R^{(j)}_{+Min}}, \Omega\right) \quad [4.15]$$

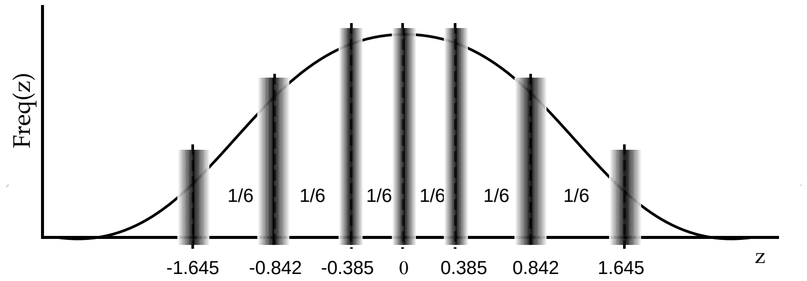
This normalization ensures that

1) Near the center of a bin, the two sigmoids which bound that bin will produce a weighting of 1 and all other sigmoids will produce a weighting of 0.

2) Near the edge of two abutting bins, only the two abutting bins will have two sigmoids with non-zero values. The weightings produces by those values will integrate to one.

In this manner, the sum of the weightings will be one for any sample receiving at least partial weighting from an interior bin. The total weighting will be one or less for any sample whose only non-zero weighting is from an end bin. Ideally all weightings will sum to one, but the weighting scheme chosen does not appear to cause any adverse effects by deviating from that ideal.

Overlapped binning can be applied in an equirange or an equiprobable binning manner. Overlapped-equiprobable binning is illustrated in Figure 4.5.



Bars indicate amount of overlap in overlapped binning with equiprobable bins
Figure 4.5

While overlapped binning results in non-integer bin values and invalidates the assumption of the bin frequencies being multinomial, I theorized that asymptotically as the number of samples goes to infinity the behavior should be χ^2.

<u>Binning & GGoF Space:</u> Binning techniques often place constraints on which μ, σ values can be evaluated without bias. For example, with discrete binning algorithms only integer μ and even, integer σ values can be evaluated without bias. Consider the situation in which μ has an integer value; a change in σ of an amount less than 2 does not affect the allocation of the samples to the bins but produces a monotonic change in the GGoF value. On the other hand, a change in σ of more than 2 changes the allocation of the samples to the bins and potentially produces a drastic non-linear change in the GGoF value. Such effects are reduced when the bins are defined using continuous weightings, i.e., overlapped binning. For this dissertation, however, the continuous nature of overlapped binning was not exploited, so every binning algorithm was applied similarly. Specifically, the GGoF values are explicitly evaluated only at integer μ and even integer σ values. Quadric B-splines [Press, Flannery et al. 1990] are used to interpolate GGoF values at intervening μ and σ values.

4.4 Accuracy and Consistency of GGoF Extrema

The χ^2_P, $\chi^2_{R\&C}$, and χ^2_{LLR} GGoF equations were evaluated using Monte Carlo simulations. These studies compared the accuracy and consistency of the strict local maxima of these functions given four different training set sizes (|*S*|=20, 40, 80, 160) from two different distributions (Gaussian and log-normal Gaussian) and using four different binning techniques (equirange, equiprobable, overlapped-equirange, overlapped-equiprobable). Accuracy is quantified by the difference between the parameters of the local maxima in GGoF and the population's ideal parameters. Consistency is revealed by the Monte Carlo one-sigma value for each parameter. Section 2.2.2 describes the method for calculating Monte Carlo one-sigma values. Chapter 2 used the one-sigma range measure to quantify the consistency of the accuracy of various classification methods. Smaller one-sigma values correspond to more consistent behavior.

Since the generalization of the GGoF functions to multivariate data is tied to the core extraction process (Section 5.2), only univariate distributions are considered. Section 4.5 summarizes the results and provides tables to simplify the comparison of the binning techniques.

4.4.1. Maximization of GGoF Functions

To more closely match the behavior of medialness functions, the GGoF functions are modified so that their maxima correspond to the best matching parameters. This is accomplished by subtracting the GGoF function values from $\chi^2_{6-1}(\alpha = 0.99) = 15.09$ and then rescaling by that value. As a result, this modified GGoF value is greater than zero for 99% of the training sets which originate from a Gaussian distribution having the corresponding μ and σ values, i.e., under the null hypothesis.

$$\chi^2_P = \left(15.09 - \sum_{i=1}^{B} \frac{\left(O^{(i)} - E^{(i)}\right)^2}{E^{(i)}} \right) / 15.09 \qquad [4.16]$$

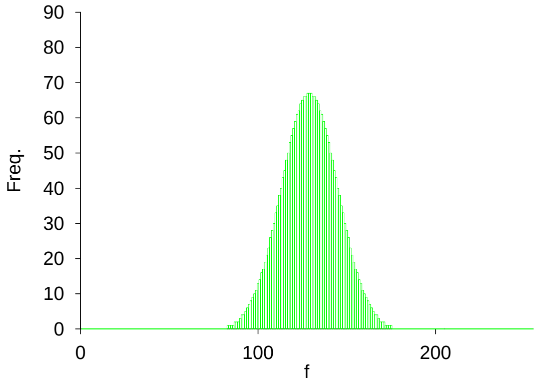
$$\chi^2_{R\&C} = \left(15.09 - \frac{9}{5} \sum_{i=1}^{B} O^{(i)} \left(\left(\frac{O^{(i)}}{E^{(i)}} \right)^{2/3} - 1 \right) \right) / 15.09 \qquad [4.17]$$

$$\chi^2_{LLR} = \left(15.09 - 2 \sum_{i=1}^{B} O^{(i)} \ \mathbf{ln} \left(\frac{O^{(i)}}{E^{(i)}} \right) \right) / 15.09 \qquad [4.18]$$

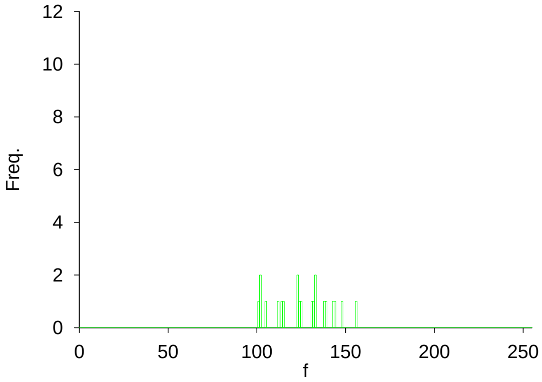
For these experiments, the local maxima were found using optimal scale surfaces [Fritsch, Pizer et al. 1994] and Brent's derivative-based line search technique [Press, Flannery et al. 1990].

4.4.2. Two Univariate Distributions

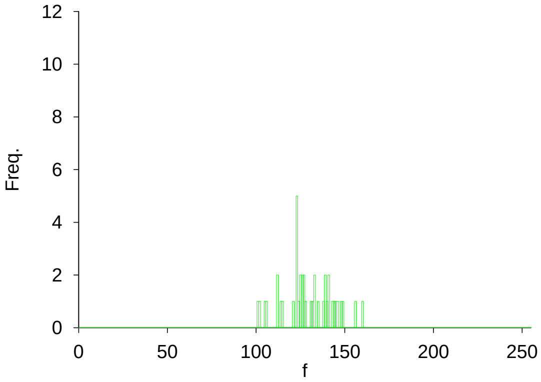
The first distribution used in the Monte Carlo studies is depicted in Figure 4.6. It is a univariate Gaussian with a mean at 128 and a standard deviation of 16. It is represented using collections of samples of size $|S|$=20, 40, 80, and 160. Examples of such sets are shown in Figures 4.7 through 4.10.



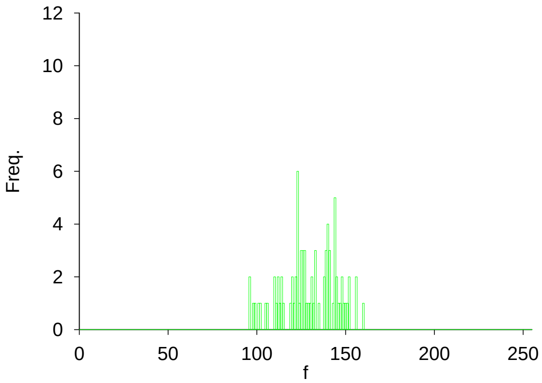
Univariate Gaussian represented using 2700 samples placed to maximize correspondence with expected frequencies
Figure 4.6



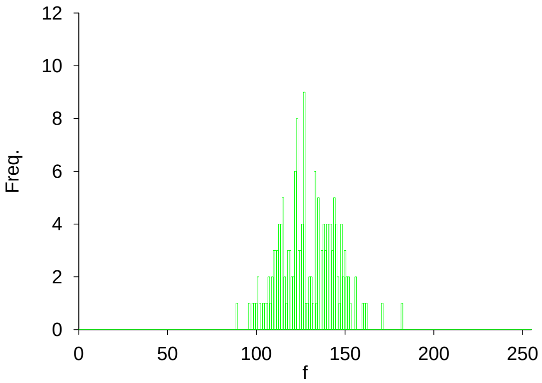
$|S|$=20 from Gaussian in Figure 4.6
Figure 4.7



$|S|$=40 from Gaussian in Figure 4.6
Figure 4.8



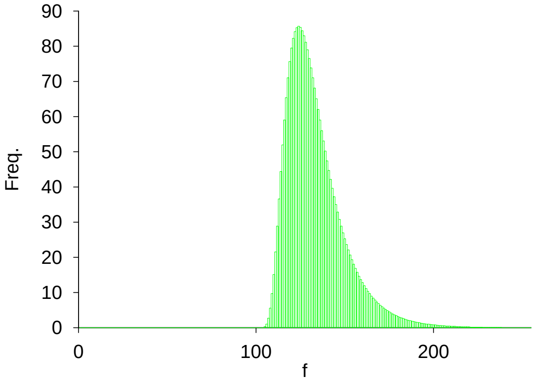
| S | =80 from Gaussian in Figure 4.6
Figure 4.9



| S | =160 from Gaussian in Figure 4.6
Figure 4.10

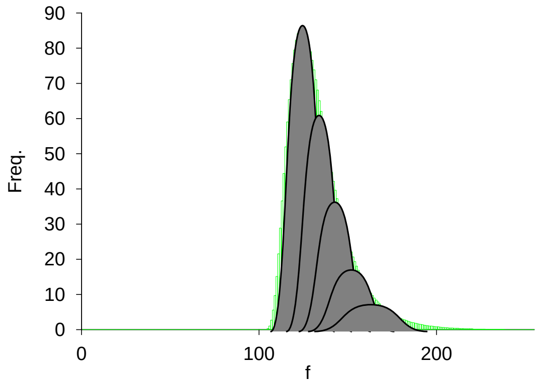
*   *   *

The second distribution is depicted in Figure 4.11. It is a univariate log-normal distribution using a log base of 1.6, μ=123 and σ=6.



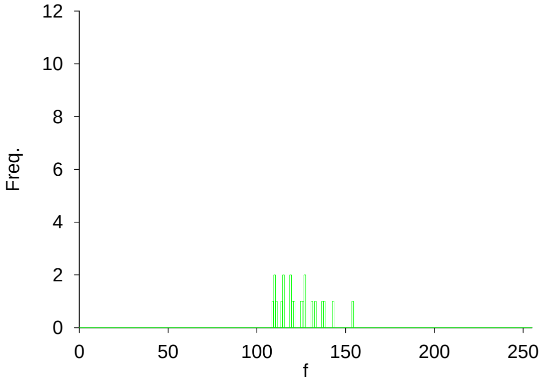
Univariate log-normal Gaussian represented via 2700 samples placed to maximize correspondence with expected frequencies
Figure 4.11

There is not a single "best" Gaussian representation of a log-normal Gaussian distribution. Instead, a log-normal Gaussian is best represented by a continuum of Gaussians, a CGMM (Fig. 4.12).

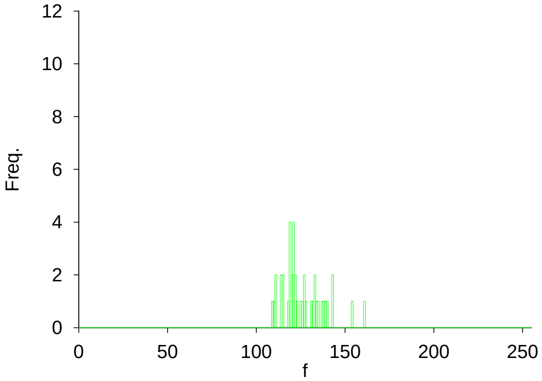


A sampled CGMM representation of a log-normal Gaussian
Figure 4.12

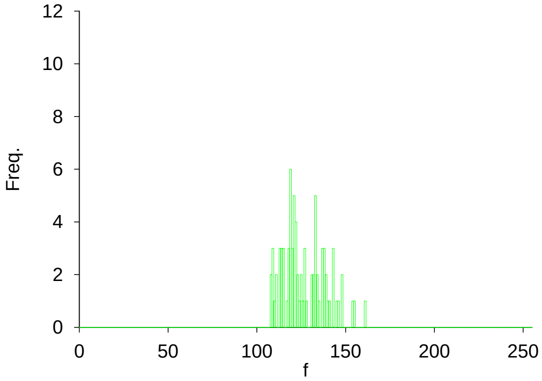
For the Monte Carlo experiments this distribution will also be represented using collections of samples of size |S| =20, 40, 80, and 160. Examples of such sets are shown in Figures 4.13 through 4.16.



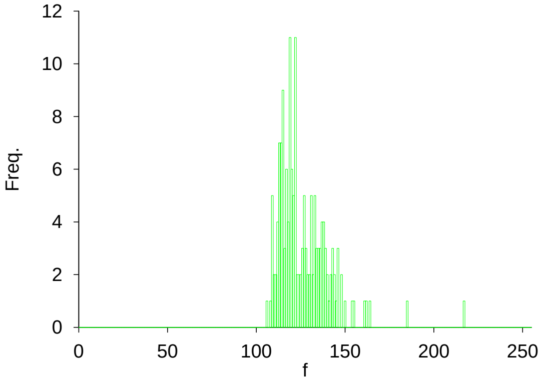
|S|=20 from log-normal Gaussian in Figure 4.12
Figure 4.13



|S|=40 from log-normal Gaussian
Figure 4.14



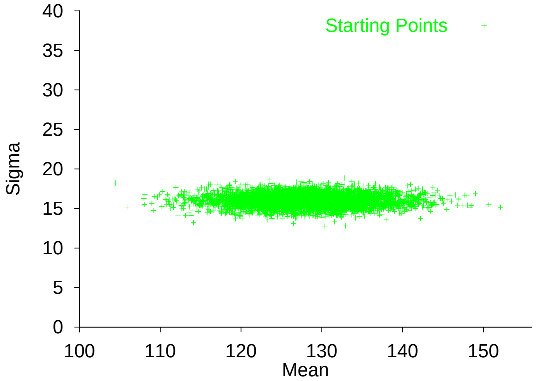
|S|=80 from log-normal Gaussian
Figure 4.15



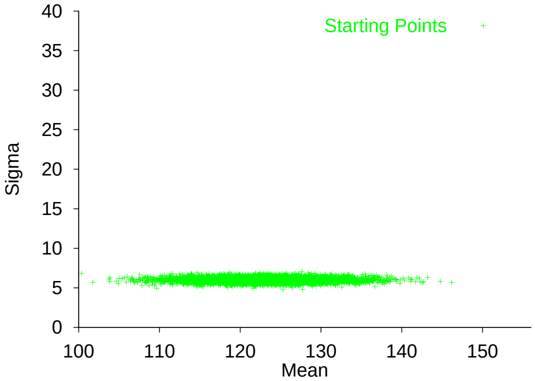
|S|=160 from log-normal Gaussian
Figure 4.16

4.4.3. Variable Starting Points

In addition to varying the size of the collection of samples used to define the GGoF space, the starting points (μ_0, σ_0) were selected from two Gaussian distributions centered at their ideal values for each sampled population and having a standard deviation of 5% of those values, i.e., μ_0=**G**(128,16) and σ_0=**G**(0.05*128=6.4, 0.05*16=0.8) for the Gaussian distribution and μ_0=**G**(123,6) and σ_0=**G**(0.05*123=6.15, 0.05*6=0.3) for the log-normal Gaussian distribution. This additional variation was motivated by the fact that for the real-world problems the starting points are estimated from the data and therefore vary. The collection of 5000 starting points used with each distribution are shown in Figure 4.17 and 4.18.



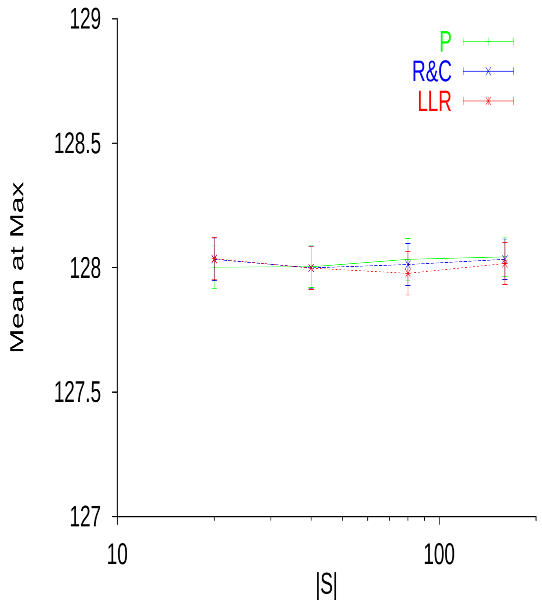
The 5000 starting points
for the Gaussian distribution
Figure 4.17



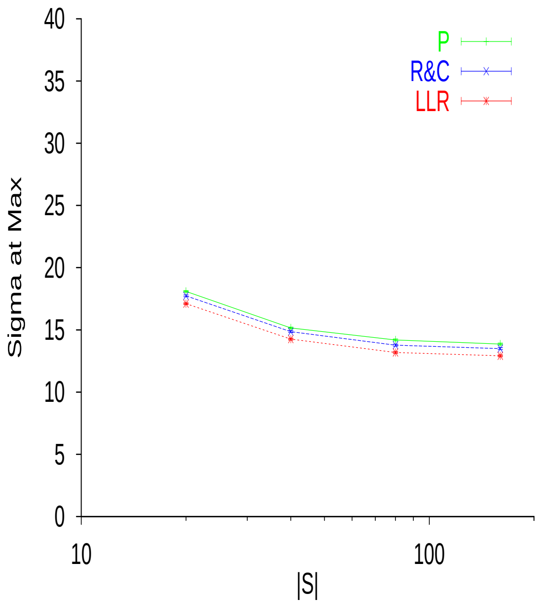
The 5000 starting points
for the log-normal Gaussian distribution
Figure 4.18

4.4.4. Equirange Binning Results

Using equirange binning and the sampled Gaussian distribution, Figures 4.19 and 4.20 illustrate the average location and the one-sigma ranges of the local maxima in GGoF for the three GGoF functions.

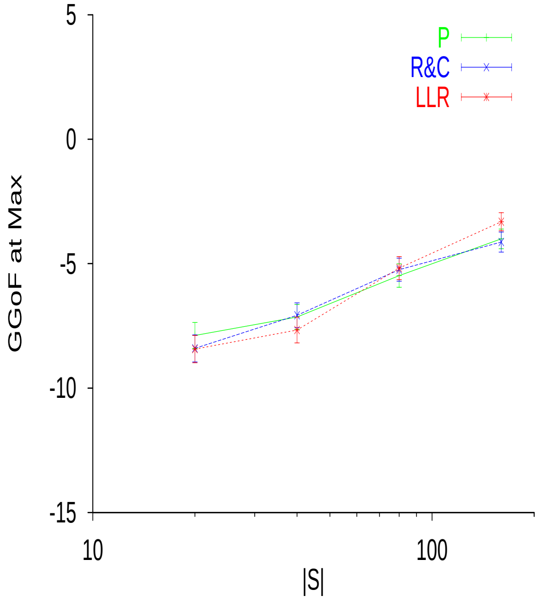


Average Mean of local max in GGoF
and one-sigma range
Figure 4.19



Average σ *of local max in GGoF*
and one-sigma range
Figure 4.20

The GGoF values at the maxima are also important. Their average values and one-sigma ranges are provided in Figure 4.21. Tables 4.1 - 4.3 list the values which are represented in these graphs.



Average GGoF value of local max in GGoF and one-sigma range
Figure 4.21

	μ		σ		GGoF	
| S |	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.0021	0.0852	18.0921	0.0709	-7.8851	0.5232
40	128.0039	0.0840	15.1517	0.0757	-7.1435	0.5033
80	128.0336	0.0836	14.1866	0.0811	-5.4787	0.4697
160	128.0437	0.0798	13.8625	0.0788	-4.0037	0.3962

X^2_P *results from Monte Carlo simulation*
Table 4.1

	μ		σ		GGoF	
| S |	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.0335	0.0855	17.7356	0.0704	-8.4043	0.5447
40	127.9995	0.0846	14.8585	0.0755	-7.0742	0.4992
80	128.0128	0.0845	13.7737	0.0820	-5.2426	0.4618
160	128.0337	0.0814	13.4942	0.0800	-4.1307	0.4021

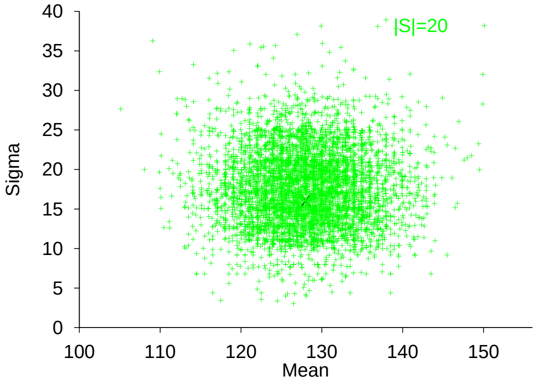
$X^2_{R\&C}$ *results from Monte Carlo simulation*
Table 4.2

	μ		σ		GGoF	
|S|	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.0361	0.0848	17.1074	0.0678	-8.4316	0.5471
40	127.9981	0.0856	14.2551	0.0744	-7.6645	0.5195
80	127.9775	0.0872	13.1854	0.0827	-5.1788	0.4629
160	128.0166	0.0843	12.9141	0.0813	-3.3135	0.3630

X^2_{LLR} *results from Monte Carlo simulation*

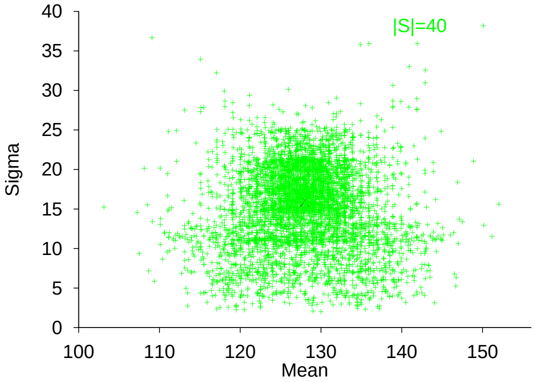
Table 4.3

These tables and graphs show an excellent correspondence between the estimated and ideal μ values. They also reveal a convergence of the GGoF values for increasing $|S|$. However, Figure 4.20 and 4.21 and Tables 4.1-4.3 reveal an unexpected asymptote for σ and unexpectedly low GGoF values at the maxima. The cause of these anomalies is revealed by scatterplots of the 5000 local maxima recorded during the Monte Carlo runs. Shown in Figures 4.22 through 4.25 are the 5000 local maxima of $\text{X}^2_{\text{R\&C}}$ function for varying $|S|$. The locations of the maxima are not unimodal for increasing $|S|$. Non-optimal local maxima are often resolved. For this GGoF function, these maxima are generally located at small scales, e.g., σ<5, in the tails of the population. Such maxima correspond to representations of outlying samples. Such consistent behavior of the non-optimal local maxima affects the asymptote for σ. Clusters of "optimal" (σ=16) local maxima are still present for the $|S|$=80 and $|S|$=160 cases. Nearly identical clusters of local maxima exist for the other GGoF functions.



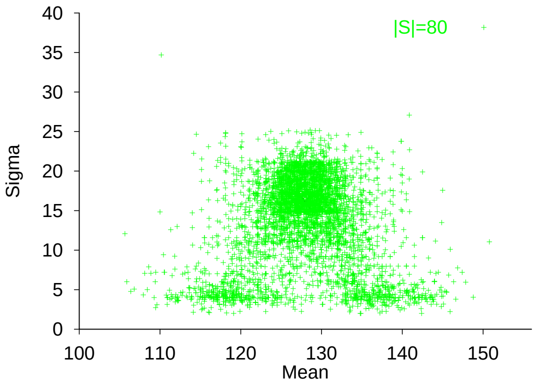
Scattergram of μ,σ of local max in GGoF using $\text{X}^2_{\text{R\&C}}$ *given 20 samples*

Figure 4.22

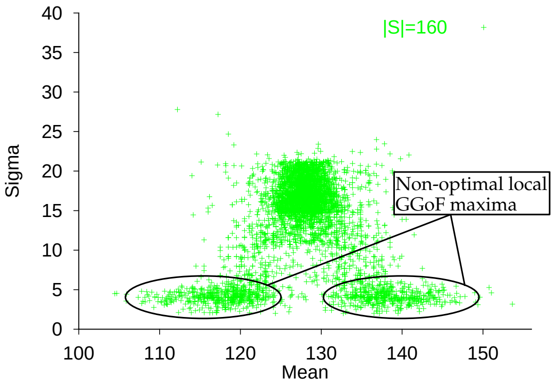


Scattergram of μ,σ of local max in GGoF using $\text{X}^2_{\text{R\&C}}$ *given 40 samples*

Figure 4.23



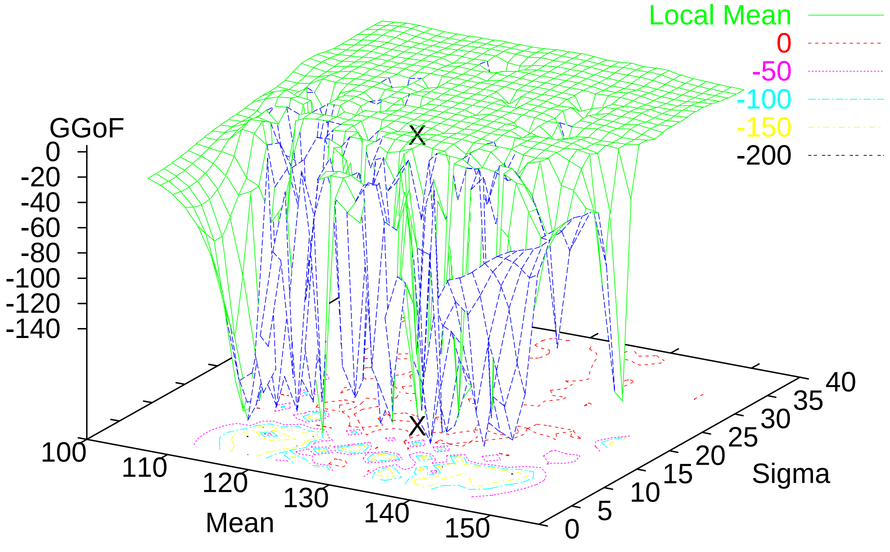
Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 80 samples
Figure 4.24



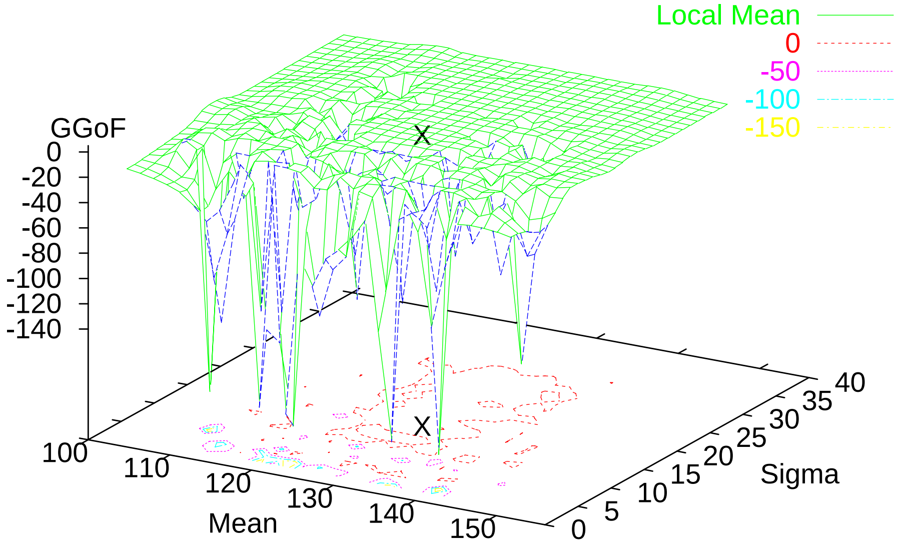
Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 160 samples
Figure 4.25

The non-optimal local maxima are also associated with poor GGoF values and therefore reduce the Monte Carlo average GGoF value. The localization of the poor GGoF values to the non-optimal local maxima is illustrated in Figures 4.26 though 4.29. The distribution of μ, σ, and GGoF values is represented by a surface whose height above a μ,σ value is equal to the L_4-norm of the GGoF values of the local maxima found near it. To aid in the visualization of this surface, its contour map is projected onto the μ,σ axis. An X has been place on the surface and contour map at the ideal μ,σ values (128,16).

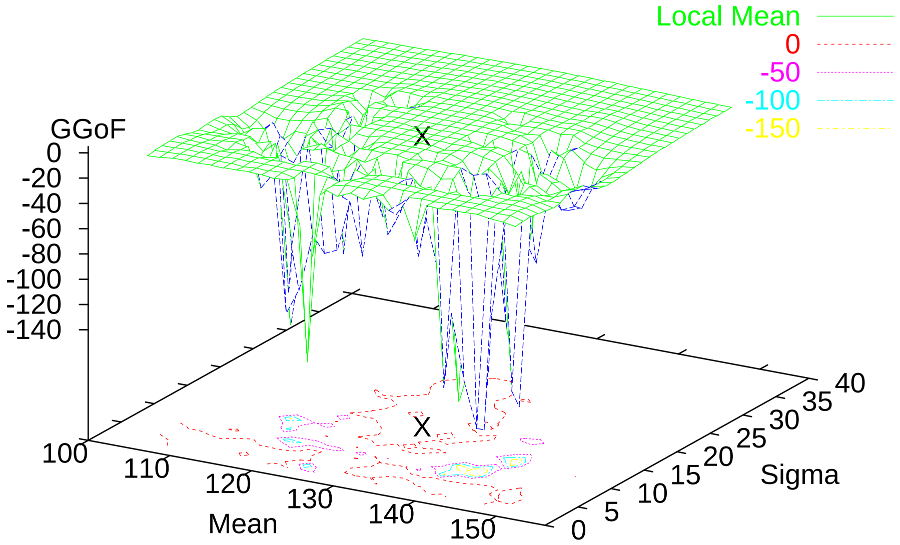
These graphs reveal the correspondence between poor GGoF values and non-optimal local maxima at small σ. Such non-optimal local maxima decrease in prevalence and become better removed from the ideal parameter values as $|S|$ increases.



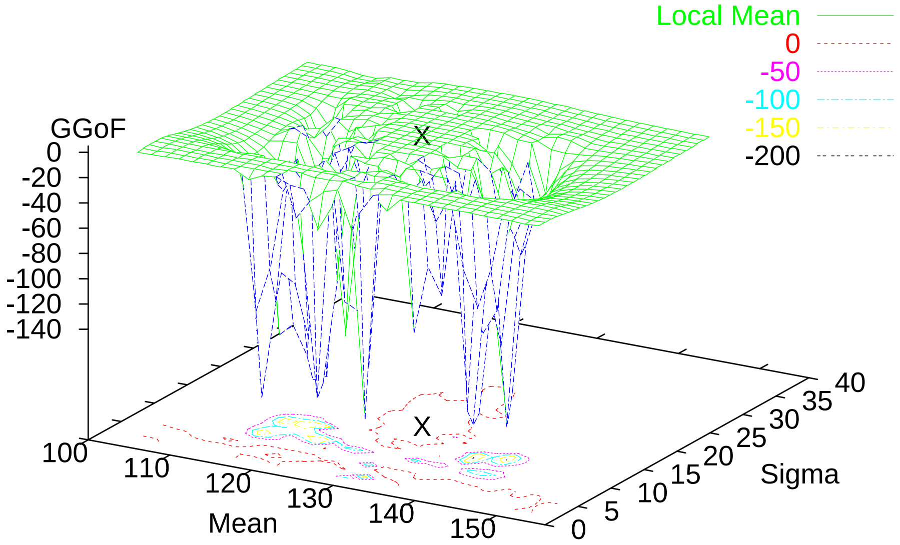
L4 Norm of GGoF at local max in GGoF
using $X^2_{R\&C}$ *given 20 samples*
Figure 4.26



L4 Norm of GGoF at local max in GGoF
using $X^2_{R\&C}$ *given 40 samples*
Figure 4.27



L4 Norm of GGoF at local max in GGoF
using $X^2_{R\&C}$ *given 80 samples*
Figure 4.28



L4 Norm of GGoF at local max in GGoF
using $X^2_{R\&C}$ *given 160 samples*
Figure 4.29

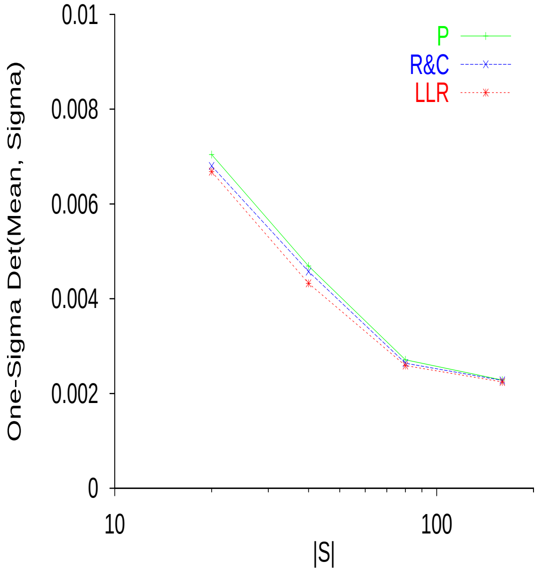
These graphs also reflect an increase in mean GGoF value at the local maxima as $|S|$ is increased. The mean GGoF value of the local maxima is greater than zero. Such correspondence is only revealed by graphs of this type. The μ, σ, and GGoF value correlations are non-linear and therefore are not revealed by a corresponding correlation matrix.

As previously stated, for each of comparison Tables 4.25-4.27 at the end of this chapter summarize the Monte Carlo statistics for each GGoF function and binning technique.

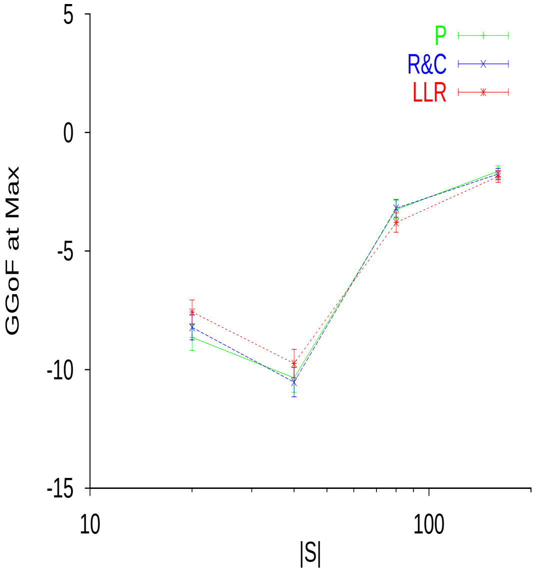
* * *

As previously stated, an "ideal" Gaussian representation for the log-normal Gaussian distribution does not exist. Graphs involving the average μ or σ values of the local maxima thus offer limited insight. The continuum of Gaussians which well represent the log-normal Gaussian, however, have a correlated spread. Ideally, the covariance of the μ and σ values about and along that spread is minimized as $|S|$ increases. Thus, the one-sigma range formed from the determinant of the covariance matrix of the μ, σ values recorded during the Monte Carlo simulation should provide meaningful information. Section 2.2.2 describes the calculation of multivariate one-sigma measures. Smaller one-sigma ranges correspond to more consistent behavior.

For the log-normal distribution and the Chi-square GGoF functions using equirange binning, the combined Monte Carlo one-sigma range for μ, σ is reported in Figure 4.30. The effect of equirange binning on the GGoF values at the local maxima given the log-normal Gaussian distributions is revealed in Figure 4.31.



One-sigma of determinant of Monte Carlo μ,σ covar.
Figure 4.30



Average GGoF of local max in GGoF and one-sigma range
Figure 4.31

	One-Sigma	GGoF	
	of		
|S|	**|Covar(m,s)|**	**Avg.**	**One-Sigma**
20	0.0066	-8.6440	0.5449
40	0.0062	-10.3474	0.6108
80	0.0047	-3.2485	0.3758
160	0.0039	-1.6307	0.2254

X_{P}^{2} *results from Monte Carlo simulation*

Table 4.4

	One-Sigma	GGoF	
	of		
|S|	**|Covar(m,s)|**	**Avg.**	**One-Sigma**
20	0.0064	-8.2185	0.5282
40	0.0057	-10.5271	0.6216
80	0.0044	-3.1970	0.3755
160	0.0038	-1.7491	0.2390

$X_{R\&C}^{2}$ *results from Monte Carlo simulation*

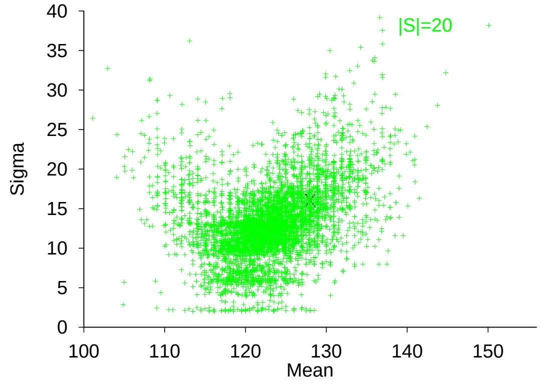
Table 4.5

	One-Sigma	GGoF	
	of		
|S|	**|Covar(m,s)|**	**Avg.**	**One-Sigma**
20	0.0062	-7.5708	0.5050
40	0.0050	-9.7440	0.6009
80	0.0041	-3.7981	0.4056
160	0.0035	-1.8580	0.2442

X_{LLR}^{2} *results from Monte Carlo simulation*

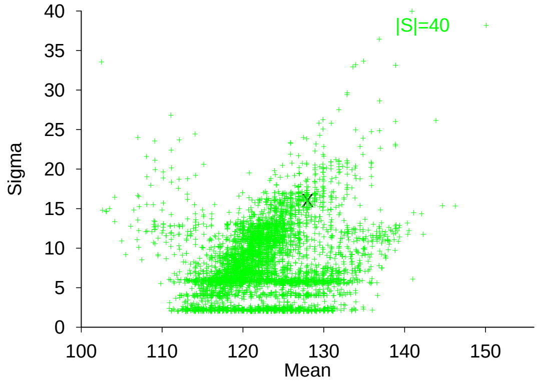
Table 4.6

Scattergrams of the μ and σ values of the local maxima for the Monte Carlo simulations are shown in Figures 4.32 and 4.33 for |S|=20 and |S|=40. They reveal the correlated spread of Gaussians which define the CGMM representation of the log-normal Gaussian.



Scattergram of μ,σ of local max in GGoF using $X_{R\&C}^{2}$ *given 20 samples*

Figure 4.32

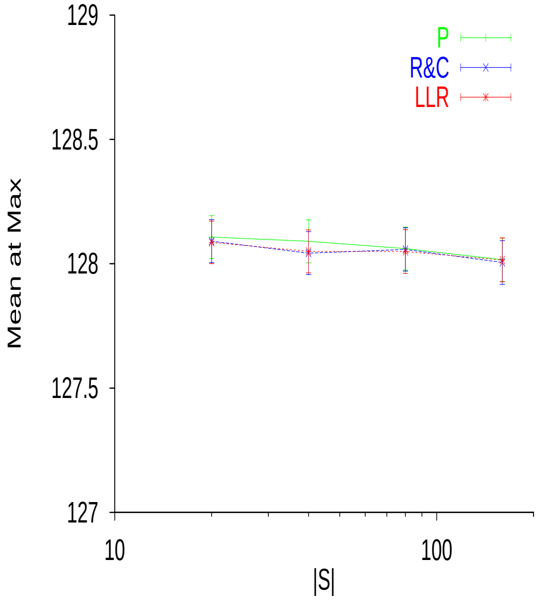


Scattergram of μ,σ of local max in GGoF using $X_{R\&C}^{2}$ *given 40 samples*

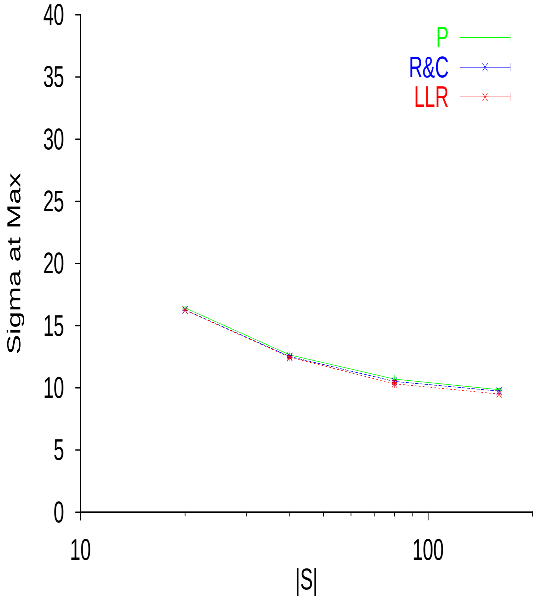
Figure 4.33

4.4.5. Equiprobable Binning Results

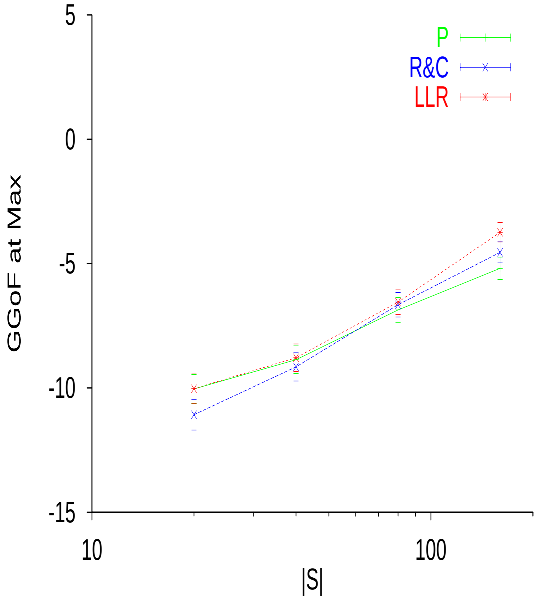
First interpretation of the theory of increased power given equiprobable binning suggests that it should offer better local maxima. This, however, seems not to be the case. Equiprobable binning appears to induce multiple, neighboring, non-optimal local maxima.



Average μ of local max in GGoF and one-sigma range
Figure 4.34



Average σ of local max in GGoF and one-sigma range
Figure 4.35



Average GGoF value of local max in GGoF and one-sigma range
Figure 4.36

Almost every one of the one-sigma ranges is larger for equiprobable binning than for equirange binning. The average GGoF values are also lower using equiprobable binning. Tables 4.25-4.27 at the end of this chapter aid in making these comparisons. Those tables summarize the Monte Carlo statistics for each GGoF function and binning technique.

	μ		**σ**		**GGoF**	
|S|	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.1071	0.0865	16.4319	0.0732	-10.0400	0.5825
40	128.0902	0.0865	12.6454	0.0814	-8.8658	0.5580
80	128.0609	0.0864	10.7077	0.0839	-6.8656	0.4974
160	128.0160	0.0873	9.8527	0.0804	-5.1896	0.4486

X^2_P *results from Monte Carlo simulation*

Table 4.7

	μ		**σ**		**GGoF**	
|S|	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.0910	0.0865	16.2709	0.0722	-11.0758	0.6158
40	128.0427	0.0865	12.5145	0.0812	-9.1536	0.5672
80	128.0573	0.0871	10.5138	0.0838	-6.6545	0.4923
160	128.0051	0.0880	9.7378	0.0803	-4.5509	0.4233

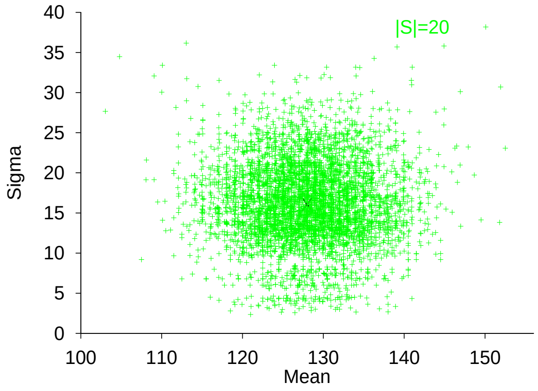
$X^2_{R\&C}$ *results from Monte Carlo simulation*

Table 4.8

	μ		**σ**		**GGoF**	
|S|	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.0859	0.0855	16.2554	0.0724	-10.0300	0.5902
40	128.0498	0.0865	12.4588	0.0816	-8.7867	0.5531
80	128.0491	0.0880	10.3299	0.0838	-6.5506	0.4930
160	128.0154	0.0884	9.4957	0.0804	-3.7419	0.3901

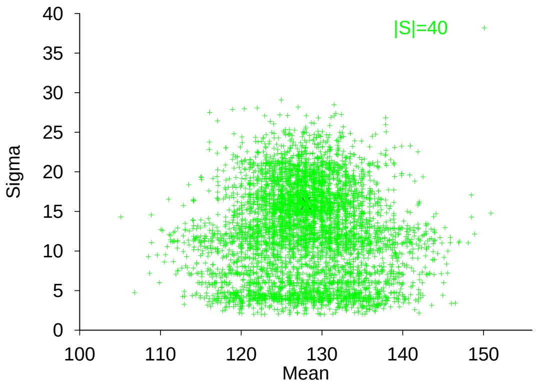
X^2_{LLR} *results from Monte Carlo simulation*

Table 4.9



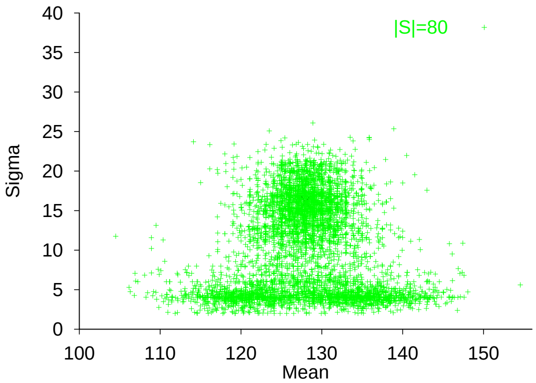
Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ *given 20 samples*

Figure 4.37

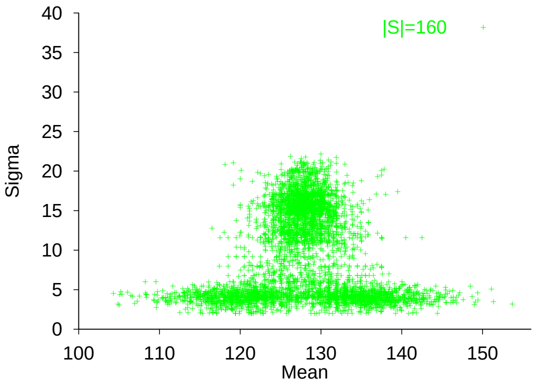


Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ *given 40 samples*

Figure 4.38



Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 80 samples
Figure 4.39

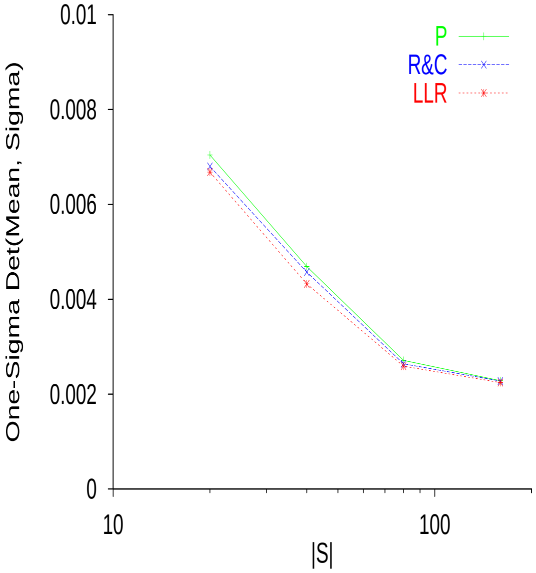


Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 160 samples
Figure 4.40

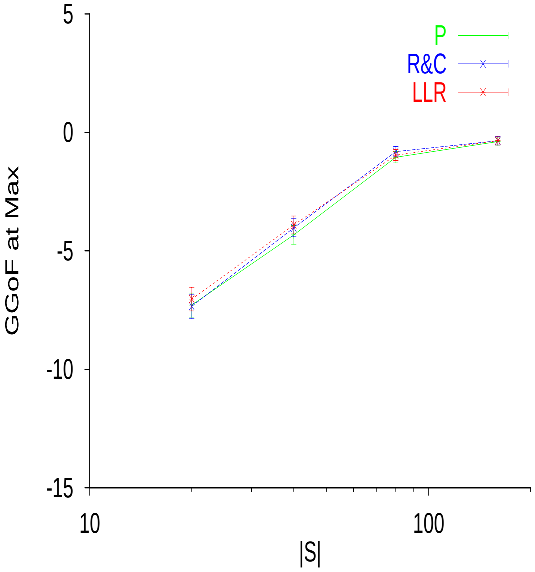
An inspection of the scatterplots of the local maxima reveals an increase in the number of non-optimal local maxima identified, c.f., Figure 4.40 and Figure 4.25. Another important difference is the location of these non-optimal local maxima. For equirange binning, they were limited to outlying μ values. For equiprobable binning, non-optimal local maxima occur even when the μ values are at the ideal!

* * *

For the log-normal Gaussian, equiprobable binning fared much better. The one-sigma ranges are consistently smaller. Also, unlike for equirange binning, the graphs are monotonic with respect to $|S|$.



One-sigma of determinant of Monte Carlo μ,σ covar.
Figure 4.41



Average GGoF of local max in GGoF and one-sigma range
Figure 4.42

	One-Sigma	**GGoF**	
	of		
$\lvert S \rvert$	**$\lvert$Covar(m,s)$\rvert$**	**Avg.**	**One-Sigma**
20	0.0070	-7.2880	0.5099
40	0.0047	-4.3236	0.3937
80	0.0027	-1.0493	0.2392
160	0.0023	-0.3864	0.1865

X^2_{P} *results from Monte Carlo simulation*

Table 4.10

	One-Sigma	**GGoF**	
	of		
$\lvert S \rvert$	**$\lvert$Covar(m,s)$\rvert$**	**Avg.**	**One-Sigma**
20	0.0068	-7.3381	0.5077
40	0.0046	-4.0255	0.3813
80	0.0026	-0.8097	0.2186
160	0.0023	-0.3501	0.1824

$X^2_{R\&C}$ *results from Monte Carlo simulation*

Table 4.11

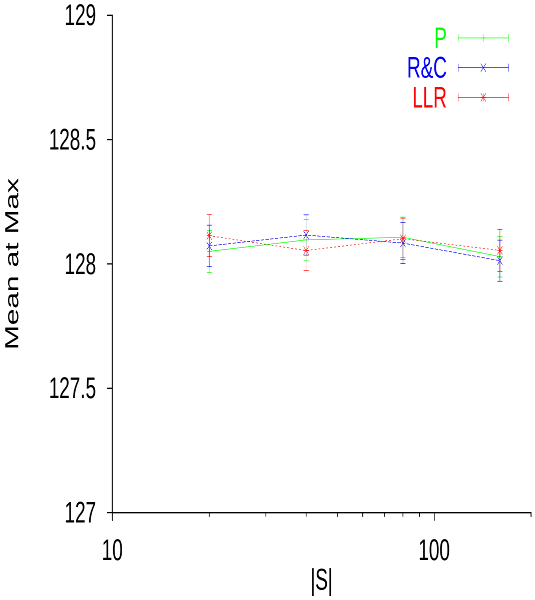
	One-Sigma	**GGoF**	
	of		
$\lvert S \rvert$	**$\lvert$Covar(m,s)$\rvert$**	**Avg.**	**One-Sigma**
20	0.0067	-7.0357	0.4999
40	0.0043	-3.9086	0.3786
80	0.0026	-0.9579	0.2325
160	0.0022	-0.3462	0.1860

X^2_{LLR} *results from Monte Carlo simulation*

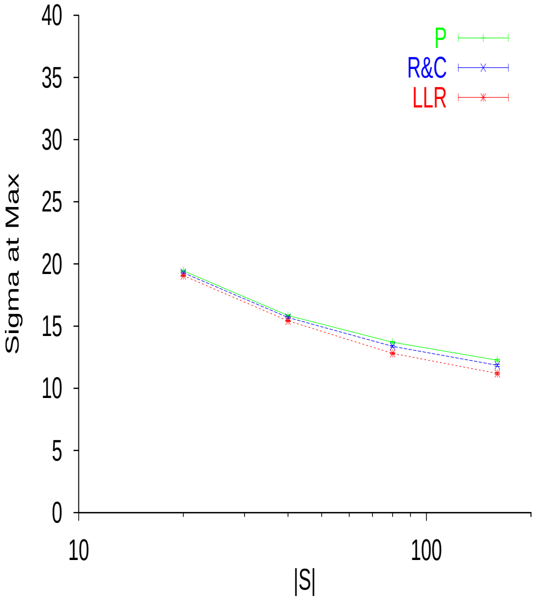
Table 4.12

4.4.6. Overlapped-equirange Binning Results

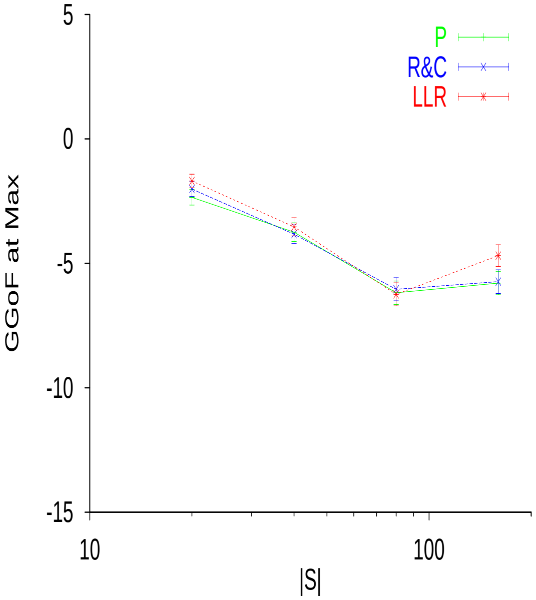
In Overlapped binning, samples are gradually transitioned from membership in one bin to membership in another for successive μ and σ values. The result is a smooth GGoF surface which allows an optimal local maxima to be more accurately and consistently found.



Average μ of local max in GGoF
and one-sigma range
Figure 4.43



Average σ of local max in GGoF
and one-sigma range
Figure 4.44



Average GGoF value of local max in GGoF
and one-sigma range
Figure 4.45

In all instances, the one-sigma values for these estimates of σ and GGoF are significantly less than those previously reported. There is not a significant change in the average or one-sigma range for the estimates of μ. Non-optimal local maxima are still resolved. They produce the same undesirable asymptotes for σ and GGoF. Tables 4.25-4.27 at the end of this chapter aid in making these comparisons. Those tables summarize the Monte Carlo statistics for each GGoF function and binning technique.

	μ		**σ**		**GGoF**	
$\lvert S\rvert$	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.0503	0.0844	19.4499	0.0586	-2.3474	0.3113
40	128.0968	0.0816	15.8521	0.0566	-3.7465	0.3734
80	128.1074	0.0812	13.7066	0.0723	-6.1812	0.4694
160	128.0294	0.0817	12.2559	0.0831	-5.7917	0.4687

$\mathrm{X}^2_{\mathrm{P}}$ *results from Monte Carlo simulation*

Table 4.13

	μ		**σ**		**GGoF**	
$\lvert S\rvert$	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.0723	0.0837	19.2835	0.0571	-2.0174	0.2946
40	128.1164	0.0810	15.6939	0.0562	-3.8296	0.3780
80	128.0839	0.0820	13.3755	0.0724	-6.0428	0.4628
160	128.0132	0.0828	11.8482	0.0840	-5.7362	0.4739

$\mathrm{X}^2_{\mathrm{R\&C}}$ *results from Monte Carlo simulation*

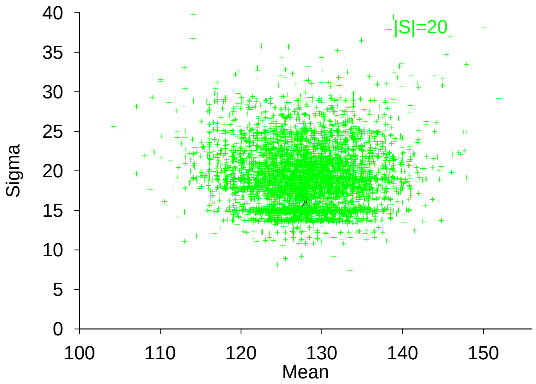
Table 4.14

	μ		**σ**		**GGoF**	
$\lvert S\rvert$	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.1140	0.0837	19.0601	0.0555	-1.6917	0.2743
40	128.0542	0.0802	15.4285	0.0544	-3.5326	0.3636
80	128.1015	0.0831	12.8055	0.0730	-6.2461	0.4635
160	128.0544	0.0846	11.1900	0.0845	-4.6897	0.4342

$\mathrm{X}^2_{\mathrm{LLR}}$ *results from Monte Carlo simulation*

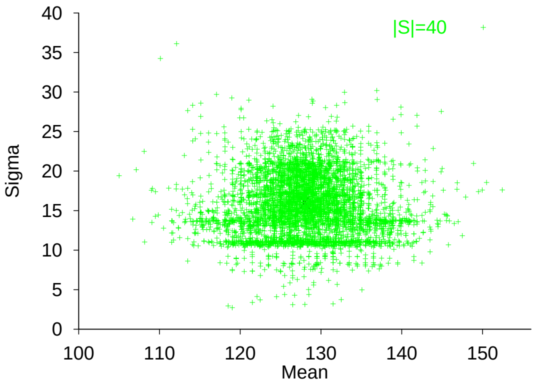
Table 4.15

When comparing the location of the non-optimal local maxima given |S|=160 (Figure 4.49) with their locations under equirange (Figure 4.40) and equiprobable (Figure 4.25) binning two things become obvious. The non-optimal local maxima are better limited to outlying μ values compared to equiprobable binning, however, they do not appear to be as well limited as with equirange binning.



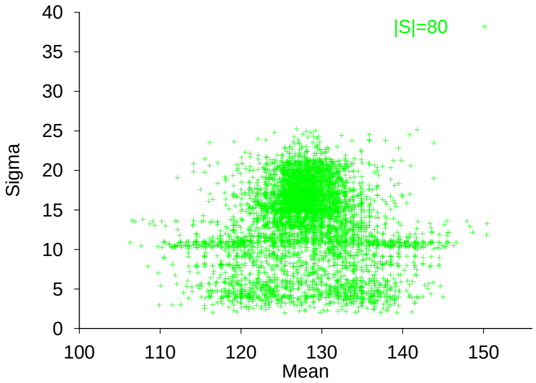
Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 20 samples

Figure 4.46



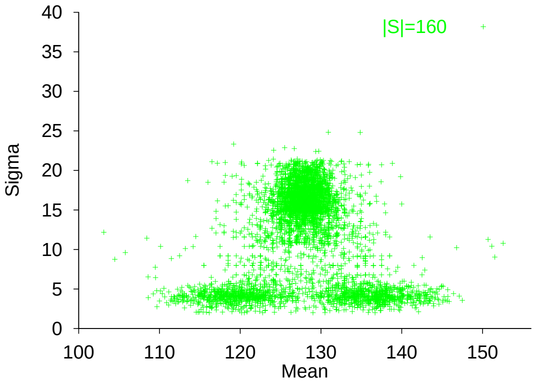
Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 40 samples

Figure 4.47



Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 80 samples

Figure 4.48

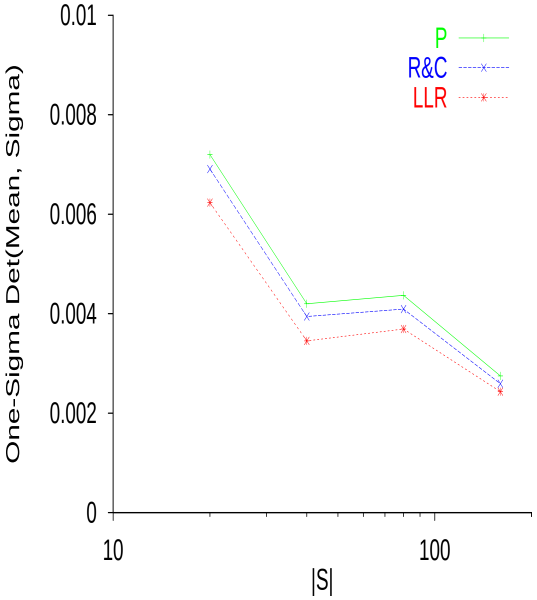


Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 160 samples

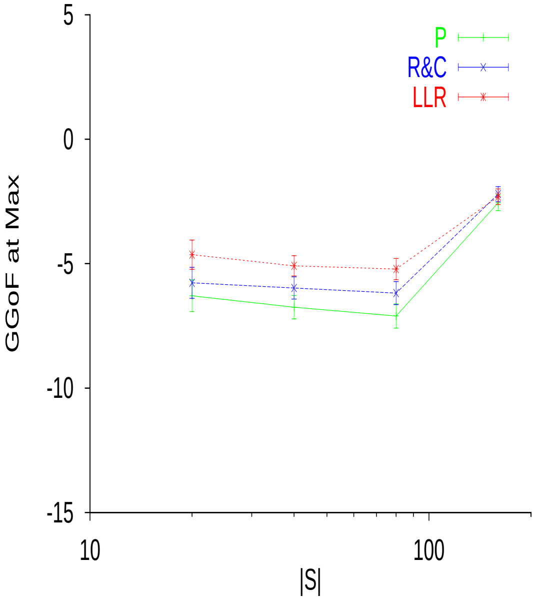
Figure 4.49

*   *   *

The effect of overlapped-equirange binning on the log-normal Gaussian distribution is most significant for the one-sigma value formed from the covariance of μ and σ. The ranges reported below are the lowest, sometimes by as much as 1/2, compared to equirange or equiprobable binning. Overlapped-equirange binning also affects the different GGoF functions differently. The X^2_{LLR} function clearly provides the best performance for all cases tested.



One-sigma of determinant of Monte Carlo μ,σ covar.
Figure 4.50



Average GGoF of local max in GGoF and one-sigma range
Figure 4.51

	One-Sigma of	**GGoF**	
|S|	**|Covar(m,s)|**	**Avg.**	**One-Sigma**
20	0.0072	-6.2908	0.6352
40	0.0042	-6.7491	0.4702
80	0.0044	-7.1035	0.4865
160	0.0027	-2.5516	0.3171

X^2_P *results from Monte Carlo simulation*
Table 4.16

	One-Sigma of	**GGoF**	
|S|	**|Covar(m,s)|**	**Avg.**	**One-Sigma**
20	0.0069	-5.7676	0.6225
40	0.0039	-5.9765	0.4442
80	0.0041	-6.1828	0.4620
160	0.0026	-2.2061	0.3029

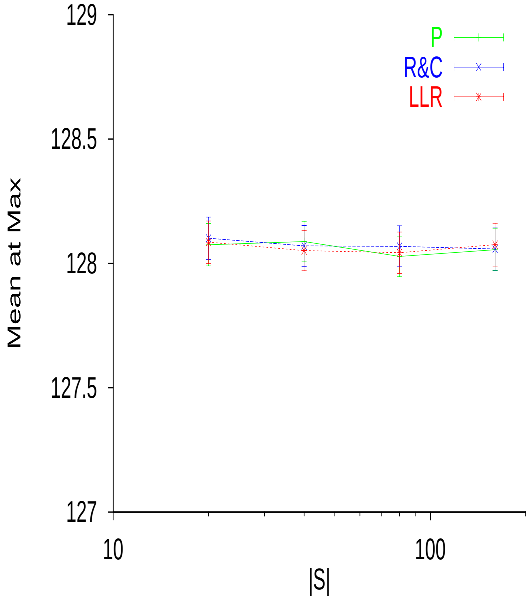
$X^2_{R\&C}$ *results from Monte Carlo simulation*
Table 4.17

	One-Sigma	GGoF	
	of		
|S|	**|Covar(m,s)|**	**Avg.**	**One-Sigma**
20	0.0062	-4.6394	0.5873
40	0.0035	-5.0866	0.4068
80	0.0037	-5.2174	0.4271
160	0.0024	-2.3078	0.3129

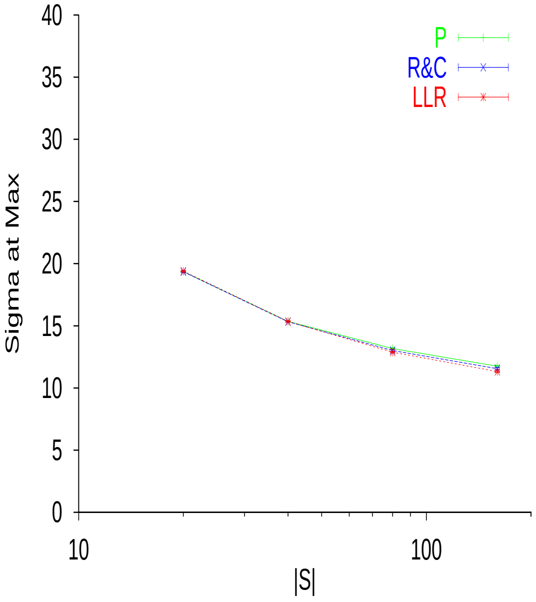
X^2_{LLR} *results from Monte Carlo simulation*
Table 4.18

4.4.7. Overlapped-Equiprobable Binning Results

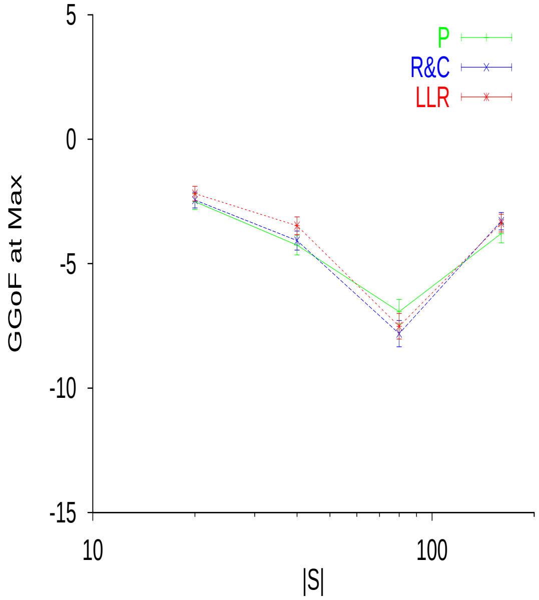
Overlapped-equiprobable binning produces the most consistent and accurate estimates of the sampled population's actual parameters. The results are given in the following figures and tables. Section 4.5 provides a summary and detailed comparison of the various binning techniques.



Average μ of local max in GGoF
and one-sigma range
Figure 4.52



Average σ of local max in GGoF
and one-sigma range
Figure 4.53



Average GGoF value of local max in GGoF and one-sigma range
Figure 4.54

The recorded one-sigma values from the σ and the GGoF estimates are significantly improved especially when using relatively few samples, i.e., | *S* | = 20 or 40. For those small sample sizes, the X^2_{LLR} function consistently provides the best performance compared to any other combination of binning technique and GGoF function. To better understand these comparisons, see Tables 4.25-4.27 at the end of this chapter. Those tables summarize the Monte Carlo statistics for each GGoF function and binning technique.

	μ		**σ**		**GGoF**	
| S |	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.0749	0.0850	19.3548	0.0576	-2.5093	0.3149
40	128.0880	0.0816	15.3573	0.0564	-4.2551	0.3954
80	128.0281	0.0818	13.1691	0.0709	-6.9306	0.4954
160	128.0551	0.0839	11.7510	0.0799	-3.7863	0.3813

X^2_P *results from Monte Carlo simulation*
Table 4.19

	μ		σ		GGoF	
|S|	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.1015	0.0851	19.3506	0.0577	-2.4454	0.3149
40	128.0705	0.0820	15.3268	0.0561	-4.0709	0.3822
80	128.0685	0.0824	13.0106	0.0708	-7.8103	0.5273
160	128.0578	0.0851	11.5473	0.0803	-3.2910	0.3522

$X^2_{R\&C}$ *results from Monte Carlo simulation*

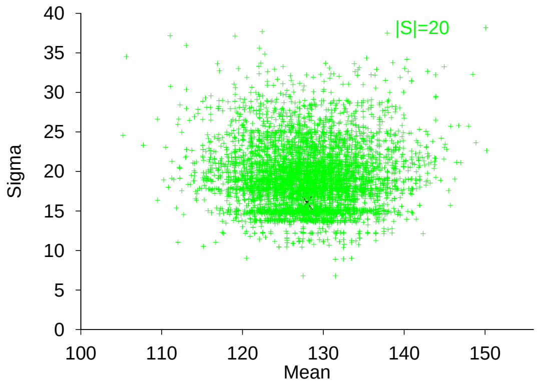
Table 4.20

	μ		σ		GGoF	
|S|	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
20	128.0853	0.0852	19.3985	0.0575	-2.1851	0.2963
40	128.0516	0.0813	15.3571	0.0567	-3.4762	0.3575
80	128.0429	0.0831	12.8740	0.0712	-7.5159	0.5159
160	128.0755	0.0863	11.3170	0.0802	-3.3717	0.3583

X^2_{LLR} *results from Monte Carlo simulation*

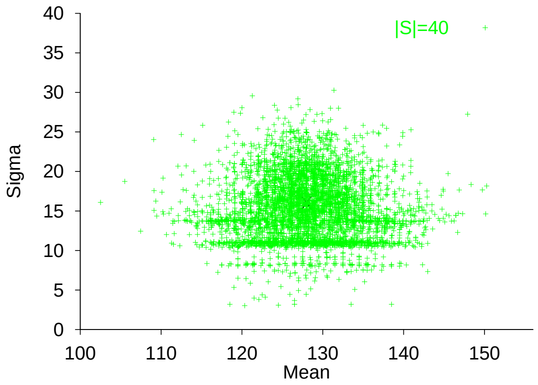
Table 4.21

Overlapped equiprobable binning provides excellent groupings of the local maxima. As with the equirange case, the limitation of non-optimal local maxima to μ values far from the ideal is clearly visible (Figure 4.57 and Figure 4.58).



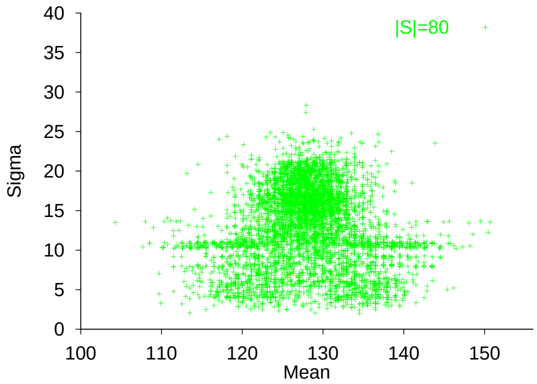
Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ *given 20 samples*

Figure 4.55

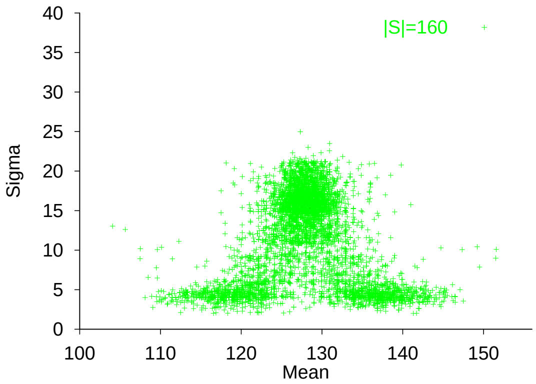


Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ *given 40 samples*

Figure 4.56



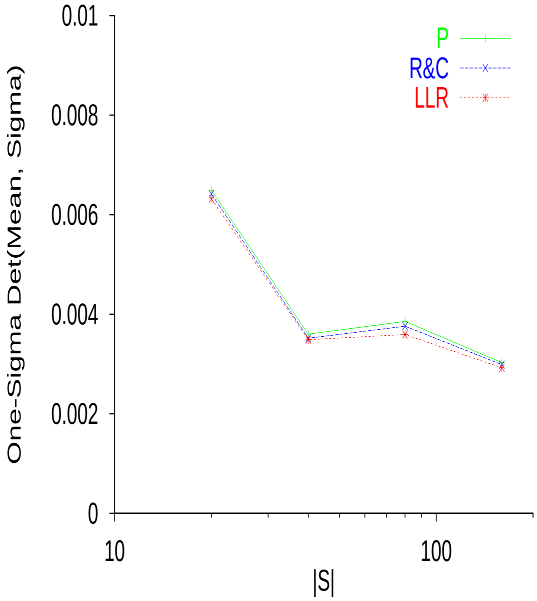
Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 80 samples
Figure 4.57



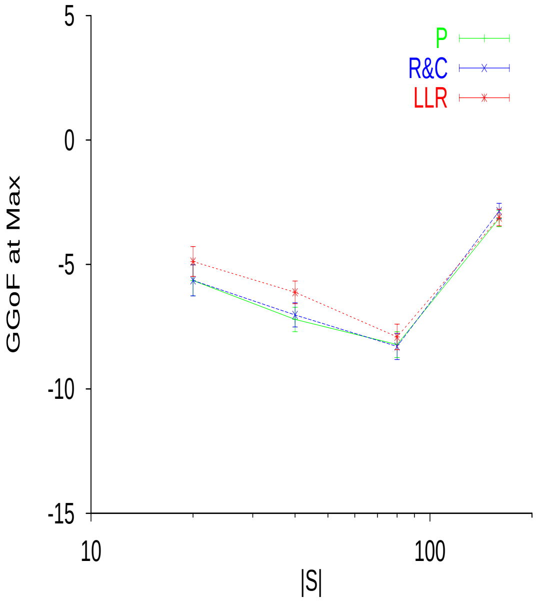
Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 160 samples
Figure 4.58

* * *

The benefit of the X^2_{LLR} function and overlapped-equiprobable binning for small samples is maintained when the log-normal Gaussian distribution is analyzed.



One-sigma of determinant of Monte Carlo μ,σ covar.
Figure 4.59



Average GGoF of local max in GGoF and one-sigma range
Figure 4.60

	One-Sigma	**GGoF**	
	of		
|S|	**|Covar(m,s)|**	**Avg.**	**One-Sigma**
20	0.0065	-5.6423	0.6250
40	0.0036	-7.2099	0.4908
80	0.0039	-8.2196	0.5203
160	0.0030	-3.1504	0.3284

χ^2_{P} *results from Monte Carlo simulation*

Table 4.22

	One-Sigma	**GGoF**	
	of		
|S|	**|Covar(m,s)|**	**Avg.**	**One-Sigma**
20	0.0064	-5.6388	0.6249
40	0.0035	-7.0284	0.4859
80	0.0038	-8.2953	0.5256
160	0.0030	-2.8551	0.3132

$\chi^2_{R\&C}$ *results from Monte Carlo simulation*

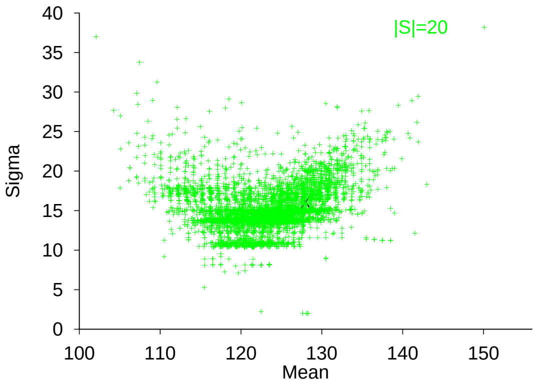
Table 4.23

	One-Sigma	**GGoF**	
	of		
|S|	**|Covar(m,s)|**	**Avg.**	**One-Sigma**
20	0.0063	-4.8825	0.5998
40	0.0035	-6.1196	0.4555
80	0.0036	-7.9107	0.5127
160	0.0029	-3.1167	0.3339

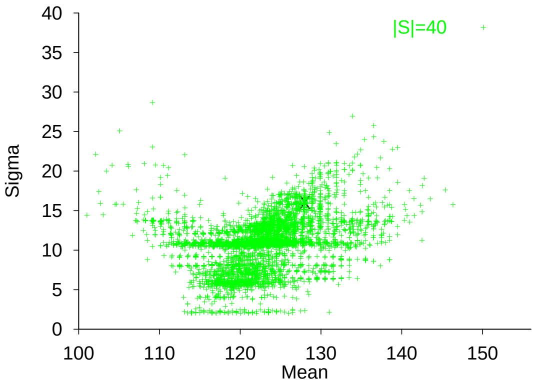
χ^2_{LLR} *results from Monte Carlo simulation*

Table 4.24

Figures 4.62 and 4.63 demonstrate the best localization of the maxima to the correlated continua of Gaussians for representing the log-normal Gaussian compared to any of the other binning techniques.



Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 80 samples
Figure 4.61



Scattergram of μ,σ of local max in GGoF using $X^2_{R\&C}$ given 160 samples
Figure 4.62

4.5 Summary

The following four tables compare the various binning techniques applied to each of the GGoF functions considered. The conclusions drawn include:

1) The technique chosen to bin the data has more influence on the accuracy and consistency of the GGoF maxima than the χ^2 function chosen to measure GGoF.

2) Overlapped-equirange and overlapped-equiprobable binning offer improvements in the accuracy and consistency of GGoF maxima.

3) Overlapped-equiprobable binning produces the most consistent estimates of σ. It has excellent performance when only a few, i.e., 20 or 40, samples are available

4) X^2_{LLR} and equiprobable binning together provide the most accurate and consistent estimates of σ.

5) The accuracy and consistency of the μ component of the maxima do not vary significantly as a function of the number of samples, the GGoF function, or the binning technique used.

6) Convergence to non-optimal local maxima can occur using any GGoF function or binning technique.

The remainder of this dissertation will focus on X^2_{LLR} using overlapped-equiprobable binning. The following tables summarize all of the Monte Carlo experiments.

$\boxed{\chi^2_P}$	μ		σ		GGoF	
|S|	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
Equi range						
20	128.0021	0.0852	18.0921	0.0709	-7.8851	0.5232
40	128.0039	0.0840	15.1517	0.0757	-7.1435	0.5033
80	128.0336	0.0836	14.1866	0.0811	-5.4787	0.4697
160	128.0437	0.0798	13.8625	0.0788	-4.0037	0.3962
Equi probable						
20	128.1071	0.0865	16.4319	0.0732	-10.0400	0.5825
40	128.0902	0.0865	12.6454	0.0814	-8.8658	0.5580
80	128.0609	0.0864	10.7077	0.0839	-6.8656	0.4974
160	128.0160	0.0873	9.8527	0.0804	-5.1896	0.4486
Olapd equirange						
20	128.0503	0.0844	19.4499	0.0586	-2.3474	0.3113
40	128.0968	0.0816	15.8521	0.0566	-3.7465	0.3734
80	128.1074	0.0812	13.7066	0.0723	-6.1812	0.4694
160	128.0294	0.0817	12.2559	0.0831	-5.7917	0.4687
Olapd equiprobable						
20	128.0749	0.0850	19.3548	0.0576	-2.5093	0.3149
40	128.0880	0.0816	15.3573	0.0564	-4.2551	0.3954
80	128.0281	0.0818	13.1691	0.0709	-6.9306	0.4954
160	128.0551	0.0839	11.7510	0.0799	-3.7863	0.3813

Summary of χ^2_P *results from various binning techniques*

Table 4.25

$\boxed{\chi^2_{R\&C}}$	μ		σ		GGoF	
|S|	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
Equi range						
20	128.0335	0.0855	17.7356	0.0704	-8.4043	0.5447
40	127.9995	0.0846	14.8585	0.0755	-7.0742	0.4992
80	128.0128	0.0845	13.7737	0.0820	-5.2426	0.4618
160	128.0337	0.0814	13.4942	0.0800	-4.1307	0.4021
Equi probable						
20	128.0910	0.0865	16.2709	0.0722	-11.0758	0.6158
40	128.0427	0.0865	12.5145	0.0812	-9.1536	0.5672
80	128.0573	0.0871	10.5138	0.0838	-6.6545	0.4923
160	128.0051	0.0880	9.7378	0.0803	-4.5509	0.4233
Olapd equirange						
20	128.0723	0.0837	19.2835	0.0571	-2.0174	0.2946
40	128.1164	0.0810	15.6939	0.0562	-3.8296	0.3780
80	128.0839	0.0820	13.3755	0.0724	-6.0428	0.4628
160	128.0132	0.0828	11.8482	0.0840	-5.7362	0.4739
Olapd equiprobable						
20	128.1015	0.0851	19.3506	0.0577	-2.4454	0.3149
40	128.0705	0.0820	15.3268	0.0561	-4.0709	0.3822
80	128.0685	0.0824	13.0106	0.0708	-7.8103	0.5273
160	128.0578	0.0851	11.5473	0.0803	-3.2910	0.3522

Summary of $\chi^2_{R\&C}$ *results from various binning techniques*

Table 4.26

$\boxed{X^2_{LLR}}$	**μ**		**σ**		**GGoF**	
|S|	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**	**Avg.**	**One-Sigma**
Equi range						
20	128.0361	0.0848	17.1074	0.0678	-8.4316	0.5471
40	127.9981	0.0856	14.2551	0.0744	-7.6645	0.5195
80	127.9775	0.0872	13.1854	0.0827	-5.1788	0.4629
160	128.0166	0.0843	12.9141	0.0813	-3.3135	0.3630
Equi probable						
20	128.0859	0.0855	16.2554	0.0724	-10.0300	0.5902
40	128.0498	0.0865	12.4588	0.0816	-8.7867	0.5531
80	128.0491	0.0880	10.3299	0.0838	-6.5506	0.4930
160	128.0154	0.0884	9.4957	0.0804	-3.7419	0.3901
Olapd equirange						
20	128.1140	0.0837	19.0601	0.0555	-1.6917	0.2743
40	128.0542	0.0802	15.4285	0.0544	-3.5326	0.3636
80	128.1015	0.0831	12.8055	0.0730	-6.2461	0.4635
160	128.0544	0.0846	11.1900	0.0845	-4.6897	0.4342
Olapd equiprobable						
20	128.0853	0.0852	19.3985	0.0575	-2.1851	0.2963
40	128.0516	0.0813	15.3571	0.0567	-3.4762	0.3575
80	128.0429	0.0831	12.8740	0.0712	-7.5159	0.5159
160	128.0755	0.0863	11.3170	0.0802	-3.3717	0.3583

Summary of X^2_{LLR} *results from various binning techniques*
Table 4.27

4.6. Bibliograph

Cressie, N. and T. R. C. Read (1984). "Multinomial goodness-of-fit tests." Journal of the Royal Statistical Society **46**(4): 440-464.

Dahiya, R. C. and J. Gurland (1973). "How many classes in the Pearson Chi-squared test?" Journal of the american statistical association **68**: 707-712.

Fritsch, D. S., S. M. Pizer, et al. (1994). Cores for Image Registration. Medical Imaging '94: Image Processing, SPIE.

Hall, P. (1985). "Tailor-made tests of goodness of fit." journal of the royal statistical society **47**: 125-131.

Ivchenko, G. I. and S. V. Tsukanov (1984). "On a new way of treating frequencies in the method of grouping observations, and the optimality of the X2 test." Soviet Mathematics Doklady **30**: 79-82.

Koziol, J. A. (1986). "Assessing multivariate normality: A compendium." Communications in Statistics: Theories and Methods **15**(9): 2763-2783.

Mann, H. B. and A. Wald (1942). "On the choice of the number of class intervals in the application of the chi-square test." Annals of Mathematical Statistics **13**: 306-317.

Pearson, K. (1894). "Contribution to the mathematical theory of evolution." Phil. Trans. Roy. Soc. A **185**: 71-110.

Press, W. H., B. P. Flannery, et al. (1990). Numerical Recipes in C. Cambridge, Cambridge University Press.

Rayner, J. C. W. and D. J. Best (1989). Smooth Tests of Goodness of Fit. Oxford, Oxford University Press.

Read, T. R. C. and N. A. C. Cressie (1988). Goodness-of-fit statistics for discrete multivariate data. New York, Springer-Verlag.

Stephens, M. A. (1974). "EDF Statistics for goodness of fit and some comparisons." Journal of the American Statistical Association **69**(347): 730-737.

Chapter 5

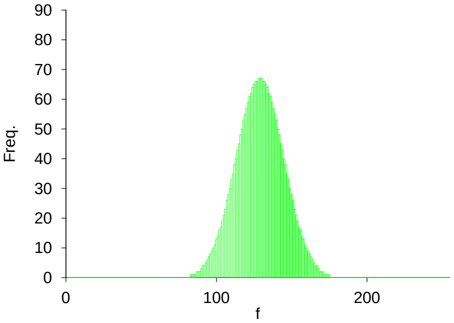
CONTINUOUS GAUSSIAN MIXTURE MODELING VIA GAUSSIAN GOODNESS-OF-FIT CORES

This chapter defines the methods of continuous Gaussian mixture modeling via Gaussian goodness-of-fit cores. The GGoF spaces of a variety of 1D distributions are illustrated. Techniques for generalizing 1D GGoF functions to ND distributions are presented and tied to the GGoF core extraction process. Specific methods for extracting GGoF cores are discussed, and the conversion of those cores to CGMM representations is detailed.

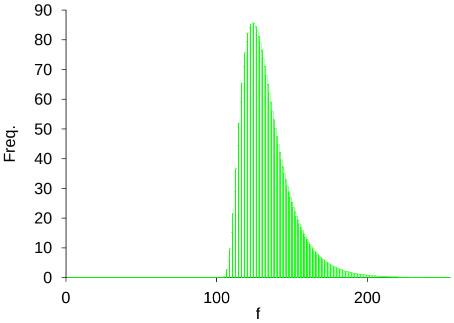
5.1. Two Dimensional GGoF Spaces of Univariate Data

The maxima of the GGoF functions analyzed in the previous chapter existed in the 2D GGoF spaces (μ,σ) of 1D distributions. These 2D GGoF spaces can be visualized to gain an understanding of the regions about their maxima.

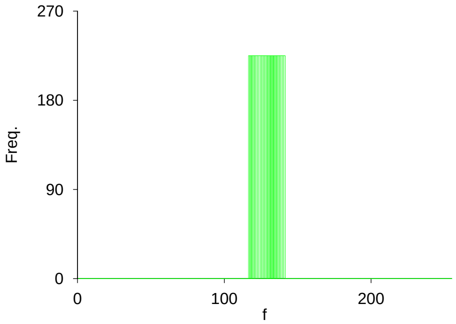
Figures 5.1-5.3 are histograms of training sets from a Gaussian (detailed in Chapter 4), a skewed Gaussian (detailed in Chapter 4), and a uniform (spanning feature values 116-140) distribution. The 2700 samples which comprise each of the training sets were allocated to feature values so that their conglomeration best represents the underlying population distribution. That is, the observed distribution is made to equal the expected distribution.



Gaussian
Figure 5.1

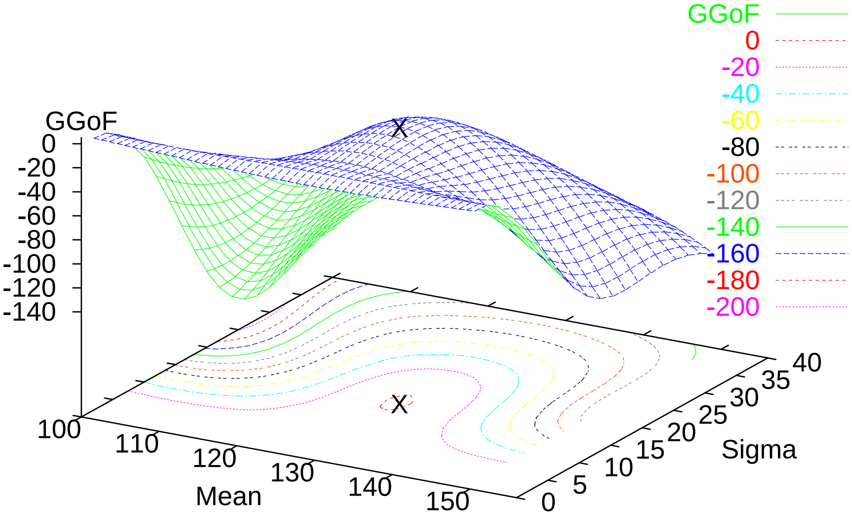


Skewed Gaussian
Figure 5.2

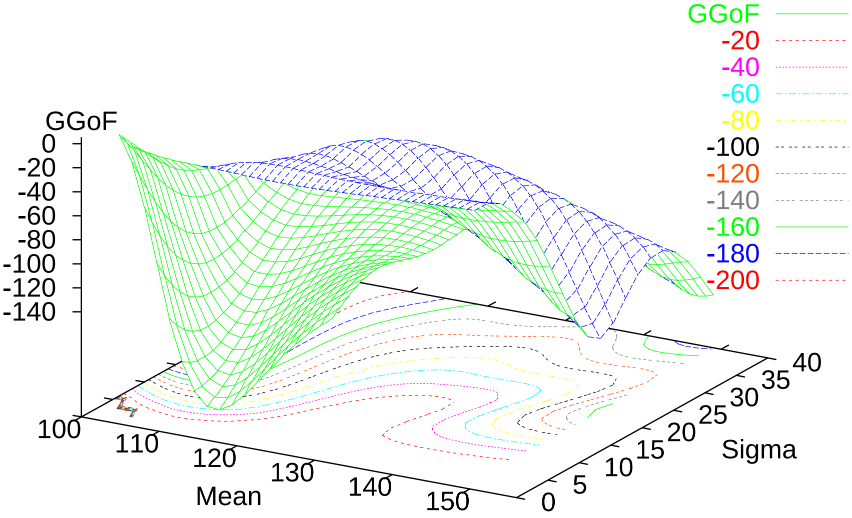


Uniform
Figure 5.3

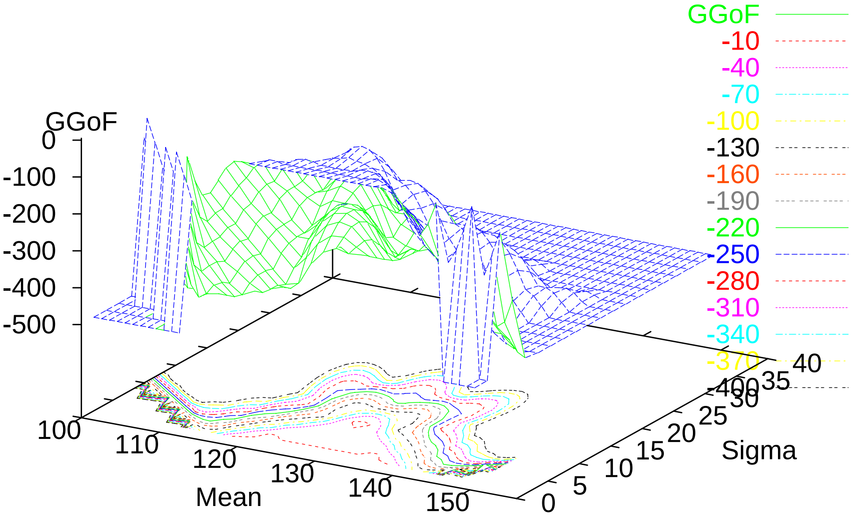
Using overlapped-equiprobable binning and X^2_{LLR}, the GGoF value at every viable combination of μ, σ can be evaluated for each data set. These values can be viewed as 2D surfaces in 3D. They are shown in Figures 5.4-5.6.



Gaussian distribution's GGoF Space depicted as a surface
X's indicate the actual parameters of the population
Figure 5.4

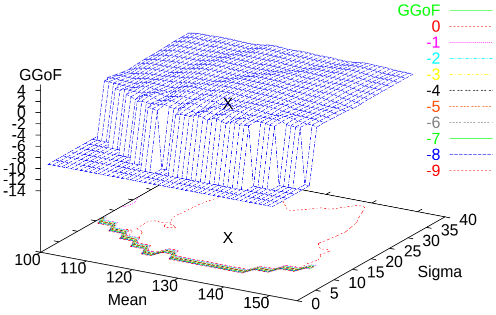


Skewed Gaussian distribution's GGoF Space depicted as a surface
Figure 5.5

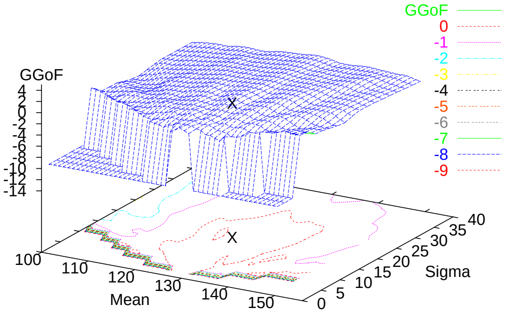


Uniform distribution's GGoF Space depicted as a surface
The GGoF values clipped at -500
Figure 5.6

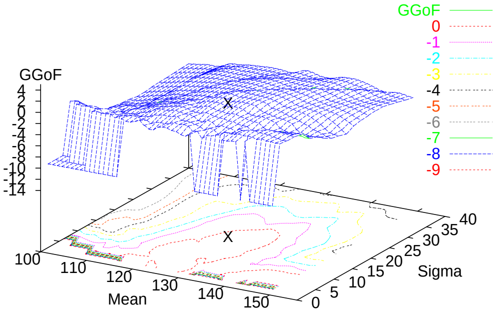
These visualizations reveal the expected extrema and the expected "smoothness" (Section 4.2) of the spaces. The GGoF spaces of the Gaussian distribution using random (not optimally distributed) samples of $|S|$=20, 40, 80 and 160 are shown in Figures 5.7-5.10. A GGoF value of -10 was assigned at points in space for which the number of local samples is less than 7, the minimum number of samples required to assure sufficient power when using equiprobable binning (Section 4.3). While the range of these surfaces is greatly reduced, they still exhibit similar shape to the ideal surface (Figure 5.4). That is, they continue to be smooth, and Chapter 4 demonstrated the correspondence of their maxima.



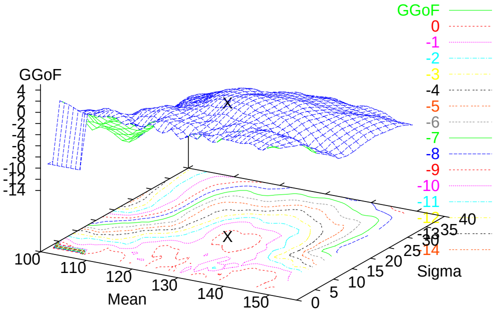
GGoF space as a surface for Gaussian distribution with |S|*=20*
Figure 5.7



GGoF space as a surface for Gaussian distribution with |S|*=40*
Figure 5.8



GGoF space as a surface for Gaussian distribution with |S|*=80*
Figure 5.9



GGoF space as a surface for Gaussian distribution with |S|*=160*
Figure 5.10

Finding the strict local maxima of GGoF for these univariate distributions, as was done in Chapter 4, is equivalent to finding 0D height ridges on these 2D surfaces. These 0D height ridges are the 0D GGoF cores of the data. Thus, the maxima analysis performed in Chapter 4 can be viewed as an analysis of the single component CGMM representations of those distributions.

The remainder of this chapter focuses on (N>1)D extruded Gaussian distributions and their (M>0)D GGoF cores.

5.2 Generalization of GGoF functions to N Dimensions

A variety of methods have been proposed for the application of 1D GGoF functions to multivariate data. This dissertation generalizes the method originally suggested by Barnett [Barnett 1976]. Specifically, the multivariate data about $\underline{\mu}$ is converted to multiple univariate distributions via projection onto each direction of a basis set (Barnett limited those directions to be the coordinate axis of feature space). The multivariate GGoF value is the average X^2_{LLR} value associated with each of those projected distributions. The set of projection directions can be defined as the eigenvectors of a projection matrix $\underline{\underline{P}}$. The eigenvalues specify the expected variance in those directions.

Define

$\underline{\mu}$ = mean being evaluated (an N-vector)
$\underline{\underline{P}}$ = projection matrix being evaluated (an NxN-matrix)
Q = Rank($\underline{\underline{P}}$) thus $Q \bullet N$
S = set of samples being evaluated
$\alpha^{(i)}$ = descending ordered eigenvalues of $\underline{\underline{P}}$, i=1..Q
$\underline{v}^{(i)}$ = corresponding eigenvectors of $\underline{\underline{P}}$, i=1..Q
s = square root of $\alpha^{(1)}$

Then the projection of the samples into each of the $\underline{v}^{(i)}$ directions is accomplished by

$$S_{\underline{v}^{(i)}} = \left\{ \underline{v}^{(i)} \underline{x}^{(j)^t} \middle| \underline{x}^{(j)} \in S \right\} \qquad [5.1]$$

and the GGoF values from each of those projections is then averaged.

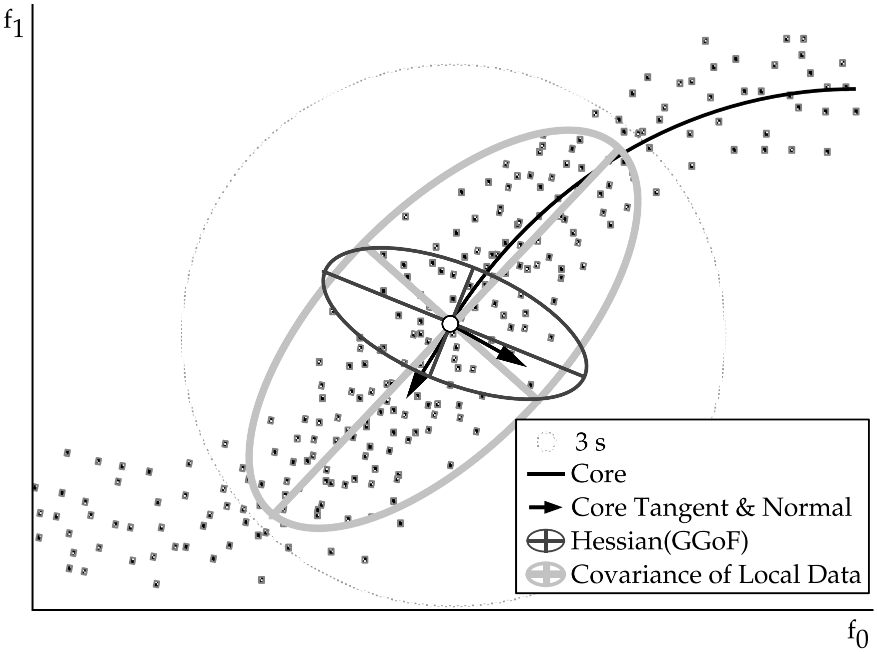
$$\mathrm{GGoF}\left(\underline{\mu}, \underline{\underline{P}}\right) = \frac{1}{Q} \sum_{i=1}^{Q} \left[X^2_{\mathrm{LLR}} \middle| \underline{v}^{(i)} \underline{\mu}^t, s, S_{\underline{v}^{(i)}} \right] \qquad [5.2]$$

Given a fixed set of data *S* and a fixed mean $\underline{\mu}$, different GGoF values will result from changes in $\underline{\underline{P}}$ even if $\underline{\underline{P}}$'s rank Q is not changed. The definition of P is critical.

As described by Equations 5.1 and 5.2, a GGoF space involves N+N(N+1)/2 parameters, a mean vector $\underline{\mu}$ and symmetric projection matrix $\underline{\underline{P}}$. It is undesirable to attempt to maximize over such a large number of parameters. As a result, the projection matrix $\underline{\underline{P}}$ is constrained to be a function of the data *S*, the mean being evaluated $\underline{\mu}$, and s the standard deviation in the maximum eigenvalued eigenvector direction of $\underline{\underline{P}}$. GGoF space is then in terms of N+1 parameters, i.e., GGoF($\underline{\mu}$,s). Whereas Barnett limited $\underline{\underline{P}}$ to be diagonal, $\underline{\underline{P}}_{ii} = s^2$ for i=1..N, the generalization of this technique to the inclusion of any set of basis directions allows the GGoF measurements to be taken with respect to the directions of the core.

5.2.1. GGoF Core Directions

The matrices generally available at a core point, $\underline{\mu}$, include the covariance of the local data $\underline{\underline{\bullet}}$, the Hessian of the GGoF function $\underline{\underline{H}}$, and the normal and tangent directions of the core. These vectors and matrices are illustrated in Figure 5.11.



Directions defined at GGoF core points
Figure 5.11

David Eberly [Eberly 1996] demonstrated that the eigenvectors of the Hessian of a medialness function closely approximate the tangents and normals of a medialness core. Thus, a medialness core can be traversed using only 2nd derivative information instead of 3rd derivative information (Section 3.4). Terry Yoo has recently proposed that statistical measures can be applied to images to approximate a variety geometric measures. This dissertation extends the ideas of Eberly and Yoo and demonstrates that the eigenvectors of the local data's covariance matrix well approximate the tangent and normal directions of a GGoF core. This allo130ithout any derivative information. Additionally, these directions are defined without having initially to assume a set of directions. For example, estimating directions via the Hessian $\underline{\underline{H}}$ of a medialness function requires calculating $\underline{\underline{H}}$, which requires the evaluation of the medialness function, which for many oriented and adaptive medialness functions requires an estimate of the core directions; yet these directions are estimated via $\underline{\underline{H}}$ which was the original goal of our calculations and therefore is undefined.

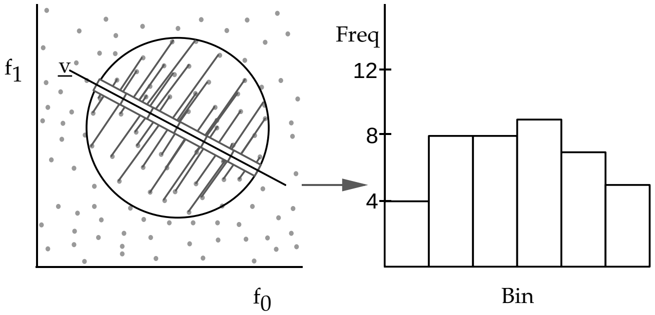
I calculate the local data's covariance matrix $\underline{\underline{\bullet}}$ using a Gaussian weighting of the covariance of the samples about $\underline{\mu}$ to provide smooth changes given small changes in location, μ, or size, s. Mathematically,

$$\Sigma_{ij} = \frac{1}{\sum_{\underline{z} \in S} G\left(\underline{z} \middle| \underline{\mu}, 3s\right)} \sum_{\underline{y} \in S} G\left(\underline{y} \middle| \underline{\mu}, 3s\right)\left(\underline{y}_i - \underline{\mu}_i\right)\left(\underline{y}_j - \underline{\mu}_j\right) \qquad [5.3]$$

where $G(\underline{y} \,|\, \underline{\mu}, 3s)$ is the value at $\underline{y}$ of a Gaussian function having a mean at $\underline{\mu}$ and a circularly symmetric standard deviation of 3s.

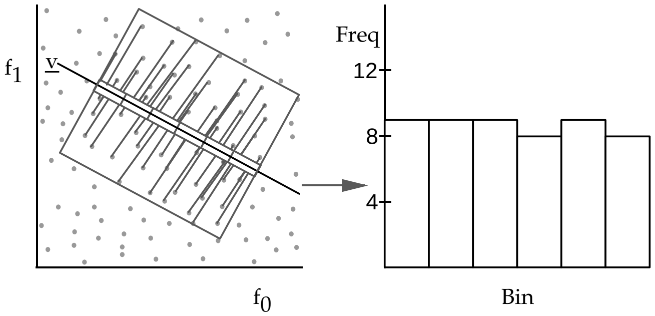
5.2.2. Projection of Local Samples

The method of projection of the multivariate data onto a single direction significantly affects the resulting univariate X^2_{LLR} value. My initial work simply projected all points within a circular region defined by the extent of the bins, e.g., 1.645s. The problem is that the amount of feature space being projected into each bin differs. For example, the projection of a multivariate uniform distribution incorrectly resembles a univariate Gaussian distribution (Figure 5.12).



Use of circular neighborhood biases binning towards a Gaussian distribution
Figure 5.12

This bias is removed by having each bin consider equal feature space areas. This is achieved by projecting the samples within a square bounding box centered at $\underline{\mu}$, spanning the extent of the bins, e.g., 1.645s, and oriented with respect to the direction of projection. As demonstrated in Figure 5.13, the resulting projection bin frequency more closely reflects that of the local distribution of samples.



Use of oriented rectangular region produces unbiased binning
Figure 5.13

This bounding box can also be made "overlapped" when overlapped binning is chosen (Section 4.3). In such situations, the samples are weighted based on their distance from the line of projection using Equation 4.12. An overlapped bounding box is used in this dissertation. It maintains the benefits associated with overlapped binning by smoothing the transition of samples into the bins as consecutive $\underline{\mu}$ and s values are tested.

5.2.3. Directions of Projection

As with medialness functions fixed, oriented, and adaptive multivariate GGoF functions can be defined. They are distinguished by which eigenvectors of the local data's covariance matrix, as the projection matrix, are considered and whether their associated eigenvalues are used.

This section illustrates the concepts it introduces using a two feature extruded Gaussian distribution. Samples are generated from this distribution using the four control Gaussians given in Table 5.1 and the steps listed in Section 2.1.1.1. The probability density function of this distribution is shown in Figure 5.14. A scattergram of the set of 2700 training samples used in the following discussion are shown in Figure 5.15.

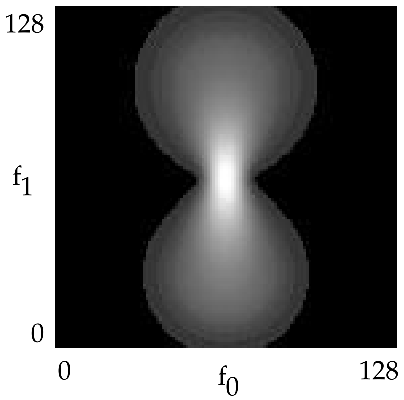
Control 0: $G^{(0)}$		**f_0**	**f_1**
Mean		26	64
	Covar	144	0
	f_0		
	f_1	0	144

Control 1: $G^{(1)}$		**f_0**	**f_1**
Mean		56	64
	Covar	4	0
	f_0		
	f_1	0	4

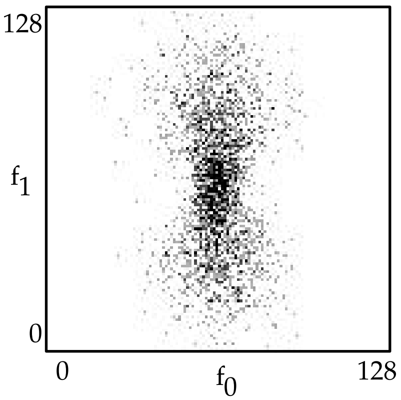
Control 2: $G^{(2)}$		**f_0**	**f_1**
Mean		72	64
	Covar	4	0
	f_0		
	f_1	0	4

Control 3: $G^{(3)}$		**f_0**	**f_1**
Mean		100	64
	Covar	196	0
	f_0		
	f_1	0	196

The parameters of the control Gaussians which are used to define Class B
Table 5.1



Probability surface of skewed Gaussian Distribution
Figure 5.14



Scattergram of the 2700 samples making the training data
Figure 5.15

This distribution is a generalized projective Gaussian distribution. It was designed to resemble the Class B distribution used in Chapter 2. Its expected GGoF core, however, is vertical, and its upper control Gaussian has a larger variance than its lower control Gaussian.

5.2.3.1. Fixed Multivariate GGoF Functions

The application of fixed multivariate GGoF functions is not affected by the local data. For example, to provide a multivariate GGoF measure equivalent to that proposed by Barnett, the direction of projection can be aligned with the coordinate axis of feature space.

$$\underline{\underline{P}}_{ij} = \begin{cases} s^2 & i = j \\ 0 & \text{otherwise} \end{cases} \qquad i=1..N,\ \ j=1..N \qquad [5.4]$$

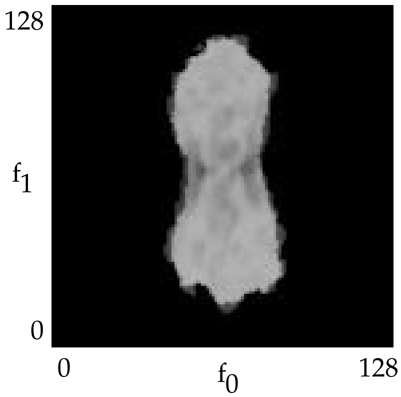
Preliminary experiments using these functions demonstrated poor GGoF core specificity. That is, for the training data in Figure 5.15, a 2D height ridge was not prominent in the 3D GGoF space. Also, given different collections of training data, if a ridge was present, it was often associated with an incorrect track through feature space and/or s.

5.2.3.2. Oriented Multivariate GGoF Functions

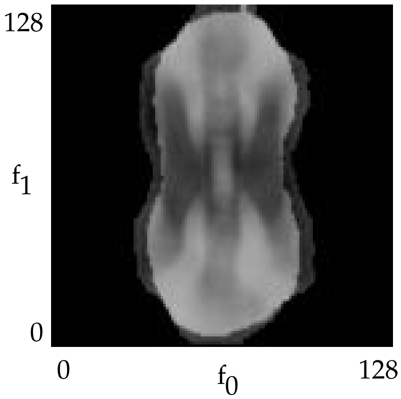
It would seem that the GGoF core's tangents and normals form a more appropriate coordinate frame than the coordinate axis. Using the core's tangents and normal directions for calculating the GGoF function values is achieved by equating the eigenvectors of the projection matrix to the eigenvectors of the local data's covariance matrix and using s^2 as the eigenvalues of the projection matrix. Such GGoF functions are called oriented-"full-rank" functions since the rank, Q, of their projection matrix equals the rank, N, of the local data's covariance matrix.

Slices through the resulting GGoF space can be viewed. Figures 5.16-5.18 show the GGoF values for various values of constant s: 4, 8, and 16. Figures 5.19-5.20 show slices through s for f_0=64 and f_1=64.

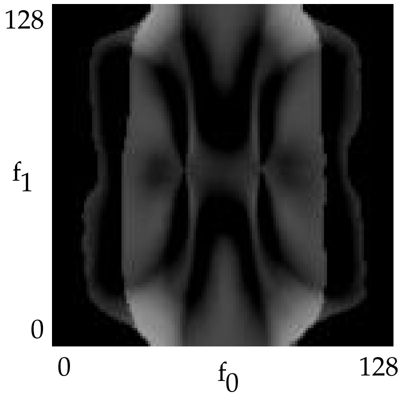
One possibly unexpected structure in GGoF space is the relatively large GGoF values at feature space locations remote to the distribution. These larger values occur because, as points farther from the distribution are tested, the number of samples within the projection bounding box begins to decrease. While the collection of samples at these remote locations are as poorly distributed as the collections of samples at points nearer the distribution, the reduced number of samples causes a drop in the expected frequency at each bin. The expected bin frequency is used to normalize X^2_{LLR} measure, and thus the GGoF values increase when the expected frequency decreases, e.g., see Equations 4.16-4.18. The Modified χ^2 measure [Read and Cressie 1988] is one of several GGoF functions which attempts to alleviate this by normalizing using the observed frequency at each bin. These outlying structures, however, will not affect the GGoF cores which exist central to the distribution.



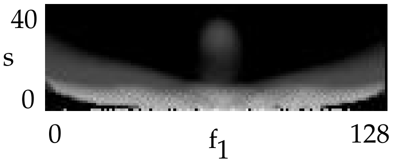
Slice through GGoF space at s=4
using a full-rank oriented GGoF Function
Figure 5.16



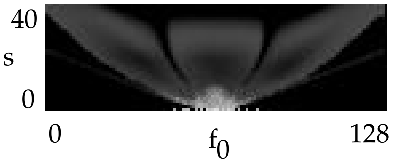
Slice through GGoF space at s=8
using a full-rank oriented GGoF Function
Figure 5.17



Slice through GGoF space at s=16
using a full-rank oriented GGoF function
Figure 5.18

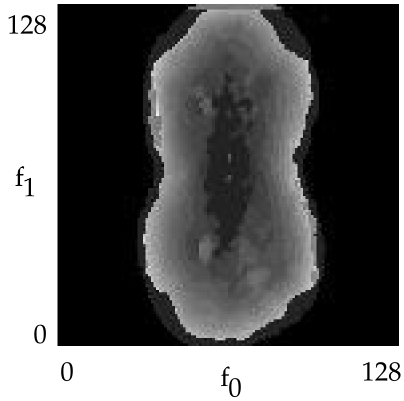


Slice through GGoF space at f_0=64
using a full-rank oriented GGoF Function
Figure 5.19

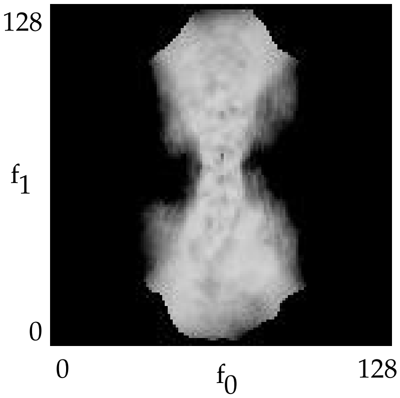


Slice through GGoF space at f_1=64
using a full-rank oriented GGoF Function
Figure 5.20

At every point in feature space, the local maxima in GGoF through s can be extracted to indicate an “optimal-s” surface. Fritsch has devised a medialness core definition based on such constructs [Fritsch, Pizer et al. 1994]. In this dissertation, however, these images are only provided for illustration; they are not being used to extract the GGoF cores. Figure 5.22 shows the GGoF values on the optimal-s surface when s=6 is used to stimulate the GGoF local maximum search through s at each feature space location. Figure 5.21 shows the s values of the GGoF local maxima of the optimal-s surface. The absence of a prominent central track in these images indicates that oriented-full-rank multivariate GGoF functions demonstrate poor GGoF core specificity.



s value of local max thru s in GGoF
given an initial s of 6
using a full-rank oriented GGoF Function
Figure 5.21

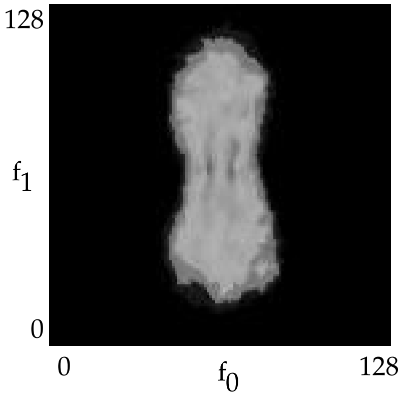


Value at local max thru s in GGoF
given an initial s of 6
using a full-rank oriented GGoF Function
Figure 5.22

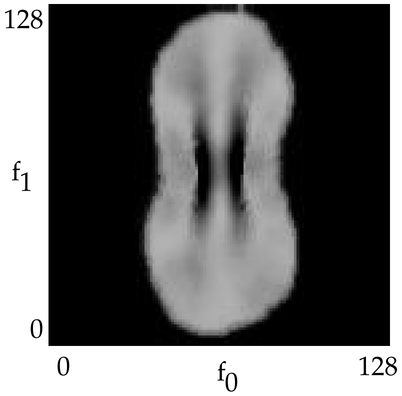
*　　*　　*

An alternative to the oriented-"full-rank" GGoF function is developed by assuming that the GGoF core captures the variations in the data along its tangent. Thus, only sample variations normal to the core need to be measured by the local GGoF function. The core's normals form the reduced rank projection matrix, Q < N. This type of GGoF function is referred to as a "oriented-normal" function. They have demonstrated excellent GGoF core specificity.

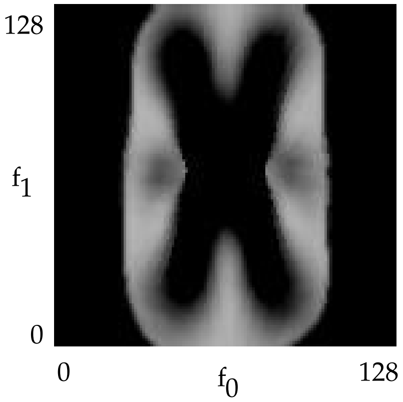
Assuming that the GGoF core's tangents are well approximated by the maximum eigenvalued eigenvectors of the local data's covariance matrix, the GGoF core's normals are well approximated by the remaining eigenvectors. Slices through the associated GGoF space for s=4, 8, and 16 are shown in Figures 5.23-5.25. Figures 5.26-5.27 show slices through s for f_0=64 and f_1=64. The GGoF values on the associated optimal-s surface developed using an initial s of 6 is shown in Figure 5.29. Figure 5.28 depicts the s values of the maxima which formed that surface. The desired 1D height ridge which is the 1D core of the data is clearly visible in all of the figures. A primary concern, however, is the existence of neighboring height ridges (see optimal-s surface, Figure 5.29). Their presence emphasizes the need for an accurate stimulation point specification process (Section 5.3.1).



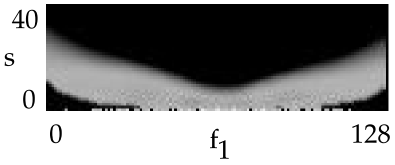
Slice through GGoF space at s=4
using an oriented-normal GGoF function
Figure 5.23



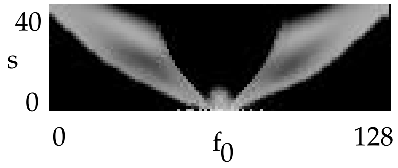
Slice through GGoF space at s=8
using an oriented-normal GGoF function
Figure 5.24



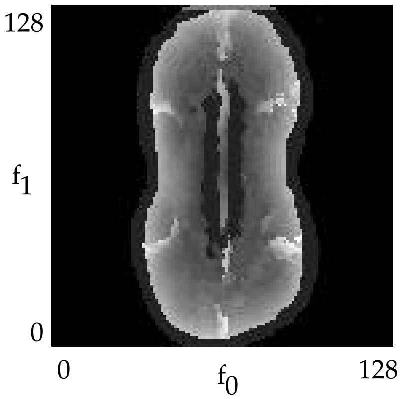
Slice through GGoF space at s=16
Figure 5.25



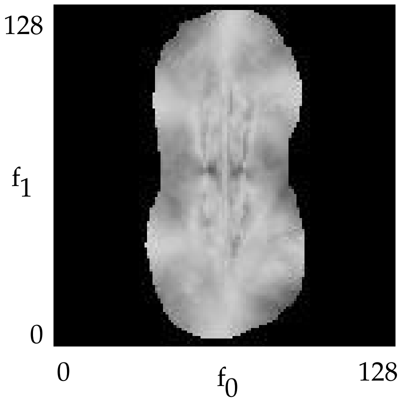
Slice through GGoF space at f_0=64
Figure 5.26



Slice through GGoF space at f_1=64
Figure 5.27



s value of local max thru s in GGoF
Figure 5.28



Value at local max thru s in GGoF
Figure 5.29

5.2.3.3. Adaptive Multivariate GGoF Functions

Adaptive GGoF functions use the eigenvalues of the local data's covariance matrix to specify the expected variance in each of the normal directions. While the oriented-normal GGoF functions are optimal for extruded Gaussians having circularly symmetric cross-sections, adaptive functions allow distributions having elliptical cross-sections to be well characterized.

As with directed GGoF functions, these GGoF functions can be of the same (Q=N) or of a reduced rank (Q<N) compared to the local data's covariance matrix.

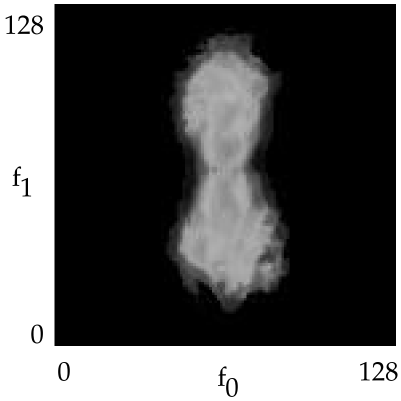
$\alpha^{(i)}(\underline{\underline{P}})$ = ascending ordered eigenvalues of $\underline{\underline{P}}$, i=1..Q
$\underline{v}^{(i)}(\underline{\underline{P}})$ = corresponding eigenvectors of $\underline{\underline{P}}$, i=1..Q
$\alpha^{(i)}(\underline{\underline{\bullet}})$ = ascending ordered eigenvalues of $\underline{\underline{\bullet}}$, i=1..N
$\underline{v}^{(i)}(\underline{\underline{\bullet}})$ = corresponding eigenvectors of $\underline{\underline{\bullet}}$, i=1..N

The Q smallest eigenvalued eigenvectors become the directions of projection

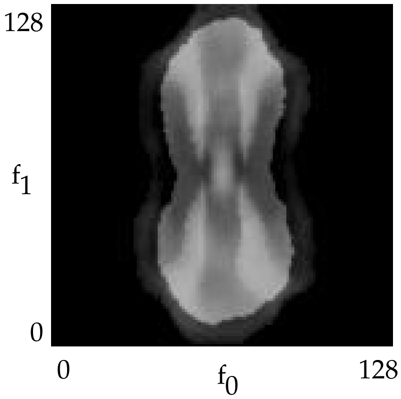
$$\underline{v}^{(i)}(\underline{\underline{P}}) = \underline{v}^{(i)}(\underline{\underline{\bullet}}) \qquad i=1..Q \qquad [5.5]$$

$$\alpha^{(i)}(\underline{\underline{P}}) = \frac{\alpha^{(i)}(\underline{\underline{\Sigma}})}{\max\limits_{j=1..Q}\left(\alpha^{(j)}(\underline{\underline{\Sigma}})\right)} s^2 \qquad i=1..Q \qquad [5.6]$$

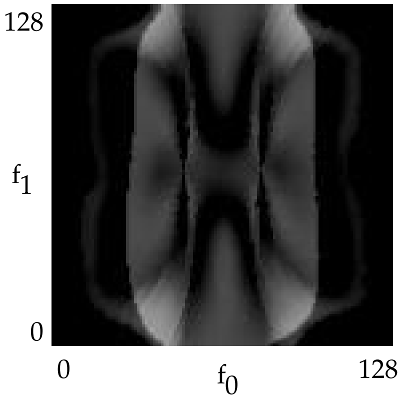
Figures 5.30-5.36 depict the slices through scale space, the GGoF values on the optimal-s surface, and the optimal-s surface's associated s values for the full-rank adaptive GGoF function (Q=N).



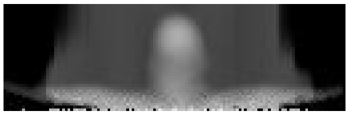
Slice through GGoF space at s=4
using a full-rank adaptive GGoF function
Figure 5.30



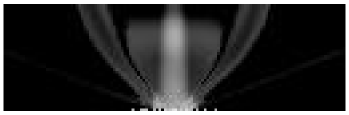
Slice through GGoF space at s=8
using a full-rank adaptive GGoF function
Figure 5.31



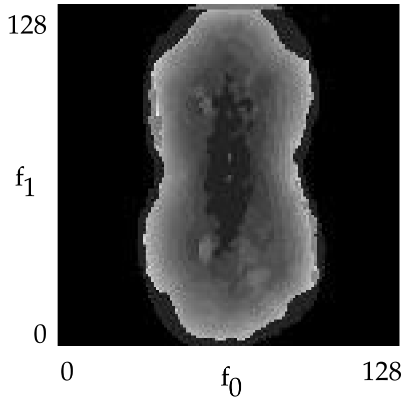
Slice through GGoF space at s=16
using a full-rank adaptive GGoF function
Figure 5.32



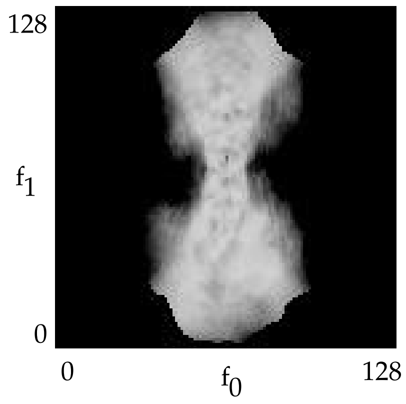
Slice through GGoF space at f_0=64
using a full-rank adaptive GGoF function
Figure 5.33



Slice through GGoF space at f_1=64
using a full-rank adaptive GGoF function
Figure 5.34



s value of local max thru s in GGoF
given an initial s of 6
using a full-rank adaptive GGoF function
Figure 5.35



Value at local max thru s in GGoF
given an initial s of 6
using a full-rank adaptive GGoF function
Figure 5.36

As with the fixed and oriented full-rank GGoF function, poor GGoF core specificity results. The consideration of the data in the tangent directions of the core serves only to degrade core specificity.

For the 2D training data being tested, adaptive normals GGoF functions provide exactly the same performance as the oriented-normal GGoF functions. Since only one direction is used for projection, the proposed adaptive normalization is inconsequential. It is expected, however, that the use of an adaptive-normal GGoF functions will provide improved core specificity in higher dimensional feature spaces because of the ability of these GGoF functions to have an elliptical shape normal to the core; fixed and oriented GGoF functions are limited to having circular shapes normal to the core. This supposition is tested in Chapter 6.

5.2.4. Summary

This section demonstrated that GGoF spaces can be generated for multivariate data. The exact nature of these spaces varies by the directions of projection used. An inspection of these spaces and their optimal-s surfaces suggests that oriented-normal GGoF functions offer the best GGoF core specificity. However, even when oriented-normal GGoF functions are used, extraneous height ridges exist on the optimal-s surfaces found, so care must be taken in the selection of the GGoF core stimulation point (Section 5.3.1).

5.3. One Dimensional GGoF Cores

The extraction of a 1D GGoF core directly follows the procedure for the extraction of a 1D medialness core. A stimulation point must be specified. A flow process finds a GGoF core point local to that stimulation point. A traversal process uses the approximate core tangent to extract the extent of that core.

5.3.1. Stimulation Point Specification

As was previously noted (Section 5.2.3.2), the existence of undesirable height ridges near the desired height ridge places importance on the stimulation point specification process. A stimulation point has two components, $\underline{\mu}_0$ and s. Three methods have been developed for specifying the feature space component, $\underline{\mu}_0$, of the stimulation point.

The first method for choosing $\underline{\mu}_0$ requires the user to completely specify $\underline{\mu}_0$. The data is displayed in N(N-1)/2 images formed from the projection of the data onto planes defined by every unique combination of coordinate axis. The user must point to a spot in at least N-1 of those projections in order to completely specify the stimulation point.

The second method is semi-automated. The user specifies a point, $\underline{x}_o$, and a neighborhood radius, r. The mean of the data within 3r of $\underline{x}_o$ becomes $\underline{\mu}_0$.

The third approach uses FGMM. It is automated yet introduces a parameter. The user must specify the number of components for FGMM (Section 2.3.7). The data is clustered using FGMM, and the component whose two nearest components are the closest (using the Euclidean distance measure) is chosen. This heuristic ensures that the chosen mean is located near the center of a densely populated region of feature space. The chosen component's mean becomes the feature space location of the stimulation point. If additional stimulation points are required for multiple core extraction, the remaining components' means can be used. For all CGMM development in this dissertation this approach is used. Also, in every example in this dissertation seven FGMM components are used. Nearly identical performance is achieved when 4 or more components are used; the number of components appears to be a non-critical parameter.

The method for specifying s_0 is paired with the method chosen for selecting $\underline{\mu}_0$. In general, this problem reduces to one of determining an appropriate neighborhood size, r_0, for calculating the local data's covariance matrix about $\underline{\mu}_0$ (Equation 5.3). Since the GGoF core is assumed to follow the M directions of the maximum eigenvalued eigenvectors of the local data's covariance matrix, s_0 is set to the (M+1)th largest eigenvalue of the local data's covariance matrix at $\underline{\mu}_0$. When methods 1 or 2 are used to specify $\underline{\mu}_0$, the user must supply r_0. When method 3 is used, r_0 is set equal to the distance between the chosen FGMM mean and its closest (measured via Euclidean distance) neighboring FGMM mean.

It will be shown in Chapter 6 that using the means of FGMM to select the feature space location and scale of the (possibly multiple) stimulation point(s) results in consistent GGoF core extractions. That is, if multiple cores are extracted from the same set of samples using the means of the FGMM components to provide the multiple stimulation points, those multiple cores generally cover the same track through GGoF space.

5.3.2. The Flow Process

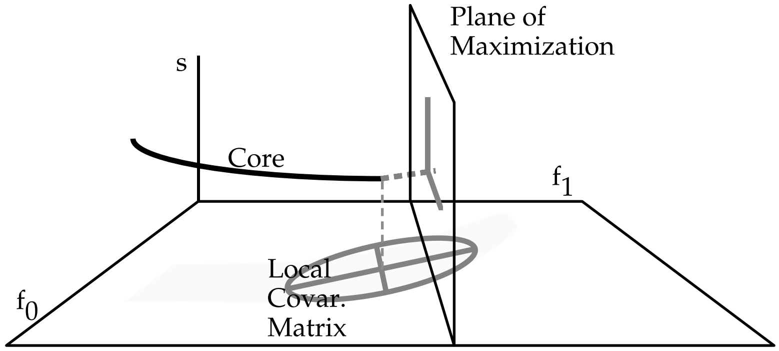
Assuming that the tangent of a core is defined by the maximum eigenvalued eigenvector of the local data's covariance matrix, the core normals are defined by 1) the remaining eigenvectors of the local data's covariance matrix and 2) a unit vector which points strictly in s. These directions define a hyperplane in GGoF space through which the local core segment passes. Dependent on the smoothness of GGoF space, by performing a gradient ascent with respect to the GGoF values on this plane, a local core point will be reached.

My approach combines the optimal-scale core definition suggested by Fritsch [Fritsch, Pizer et al. 1994] and the height ridge core definition developed by Eberly [Eberly, Gardner et al. 1994]. Both my approach and Fritsch's maintains s as a direction of maximization. My approach, however, does not independently maximize through s and then attempt to find ridges as did Fritsch's optimal-scale medialness extraction process. My approach's gradient ascent process maximizes in all relevant directions simultaneously. My approach is also like Eberly's in that it finds the local maximum in a core normal plane. However, in contrast to the practice of Eberly, using the eigenvectors of the Hessian of medialness space, the eigenvectors of the covariance matrix have no s (scale) component. A vector of pure s must therefore be added to form a proper basis in GGoF space.

For this dissertation the gradient ascent search is implemented using Brent's line search technique [Press, Flannery et al. 1990]. Given a starting point in GGoF space, the projection of its gradient onto the plane is calculated, and a line search is performed along the resulting direction. From that new maxima the process is repeated until the projection of the gradient onto the plane is near zero, e.g., less than 0.001 of its total magnitude.

5.3.3. The GGoF Core Traversal Process

The GGoF core traversal process is performed in the positive and negative tangent directions from the initial core point. The traversal process consists of two simple steps. These steps are illustrated in Figure 5.37.



Local Covar matrix defines core tangent and the plane in which the next core point is a local max.
Figure 5.37

First, the current core point's local covariance matrix is calculated. It's eigenvectors are searched for the one whose dot product with the previous core point's tangent eigenvector is of maximum magnitude. The chosen eigenvector is generally the eigenvector with the largest eigenvalue; at the initial core point the tangent direction is assumed to be best approximated by the eigenvector with the largest eigenvalue. In regions of feature space having sparse data or a nearly circularly symmetric local distribution, however, the eigenvalues/eigenvectors may "swap." This heuristic reduces the ill effects of such swapping. To maintain a consistent direction of traversal, if the sign of the chosen dot product is negative, the corresponding eigenvector is negated. The resulting vector approximates the core tangent.

Second, a step of 0.1 feature space units is taken along the approximate tangent direction from the current core point. A flow process is then initiated within the plane defined by the core's normals using an initial s value equal to the previous core point's s value.

These steps are repeated until a core termination criterion is met.

5.3.4. GGoF Core Termination and Recovery Criterion

A variety of heuristics were evaluated for the termination of GGoF cores. As with medialness cores, GGoF core termination is still an open problem. Aside from the standard criterion of experiencing "too large" of a change in traversal direction, the GGoF values of the core provide an excellent termination criterion which is not available to medialness cores. Empirical evidence suggests that encountering a GGoF value of -10 or less is a suitable stopping criterion. This single criterion is used for the termination of all of the GGoF cores extracted for this dissertation.

Reliance on the GGoF value for termination allows the rate of core change to be used for core recovery. The training sets used in this dissertation generally contained only 900 samples spanning a 256x256 region. Experience indicates that this corresponds to a "high noise" environment from which to extract GGoF cores. To add insensitivity to the core traversal process, the rate of change in the tangent of the core is used to identify suspect core points and to halt their inclusion into the core without causing termination of the traversal process. Such points are "stepped over" using the tangents of the previous valid core point. Even though various heuristics and interpolations could be used to back-fill the suspect points, such techniques have proven to be unnecessary and undesirable considering their generally ad hoc nature.

5.4. (M<N) Dimensional GGoF Core Extraction

To extend the above process to MD GGoF cores, only the traversal process needs to be modified.

For MD cores, an MD tangent frame needs to be tracked. These directions are approximated using the local data's covariance matrix. Its eigenvectors are ordered using the eigenvectors of the previously encountered core point. The signs of the associated dot products determines any negation needed to ensure a consistent direction of traversal. Normal planes with respect to each of the tangent directions are search for connected core points. The main problem with this approach is the significant amount of memory required. In its current implementation, only extremely small cores can be extracted. As a result, the extraction of these cores will not be demonstrated in this dissertation.

5.5. (M=N) Dimensional GGoF Core Extraction

When the dimensionality of the GGoF core equals the dimensionality of feature space, the implied assumption is that the distribution is not an extruded Gaussian. Such problems are outside of the stated aims of this dissertation. Some tests, however, have been performed, and the results are encouraging.

When M=N, a simple connected components method can be used to extract the GGoF core. Specifically, all of points bordering the current feature space extent of the core are evaluated and added to the core if a core termination criterion is not met. That is, at every neighboring feature space location the maxima in GGoF through s is calculated; using the GGoF criterion those points whose GGoF values are sufficient are added to the core; their neighbors are then tested.

Because of their lack of correspondence with the problem set of this dissertation, the extraction of (M=N)-D cores will not be evaluated in this dissertation.

5.6. Variations on GGoF Core Extraction

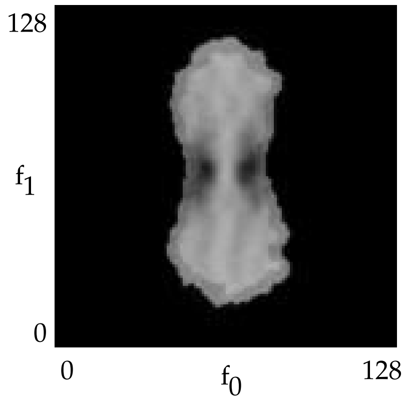
The analysis of the GGoF spaces and the GGoF core extraction process lead to the development and investigation of several variants to the core extraction process. Two of the most promising of such modifications focus on the evaluation of off-core points and alternative methods for interpolating GGoF space values and derivatives.

5.6.1. Directed GGoF functions and Off-Core Points

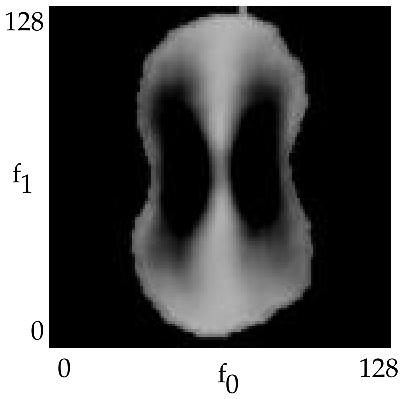
It is possible to calculate the local data's covariance matrix at every point in the GGoF space and thereby specify a unique projection matrix for every point in GGoF space including "off-core" points. Improved core specificity, however, is achieved by limiting the projection directions of off-core points to the projection directions of the neighboring core points. Off-core points are used by the approximation/interpolation technique to provide values at non-integer GGoF space locations (see section 4.3). Constraining the directions of projection in this manner improves core specificity.

For the training data in Figure 5.15, since the expected core direction is strictly vertical, the direction of projection, the core normal, can be assumed to be horizontal throughout feature space. The resulting oriented-normal GGoF space is depicted in the slices in Figures 5.38 through 5.42. The corresponding optimal-s surface and its s values are shown in Figures 5.43 and 5.44.

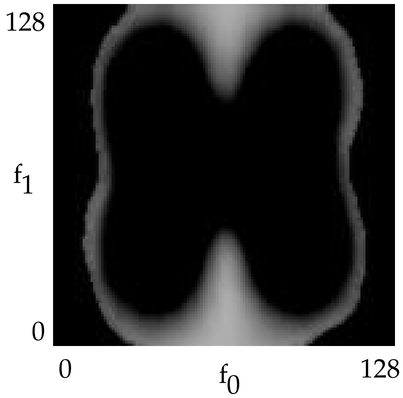
Throughout the remainder of this dissertation the normal directions of the nearest core point will be used in the evaluation of each off-core.



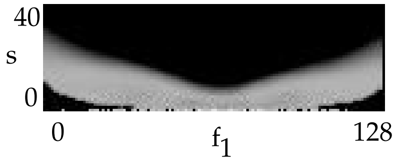
Slice through GGoF space at s=4
using an oriented normal GGoF function
Figure 5.38



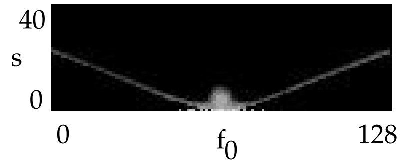
Slice through GGoF space at s=8
using an oriented normal GGoF function
Figure 5.39



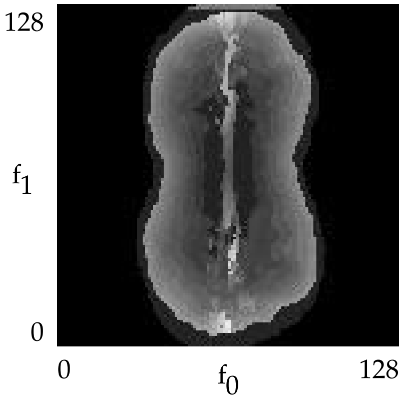
Slice through GGoF space at s=16
using an oriented normal GGoF function
Figure 5.40



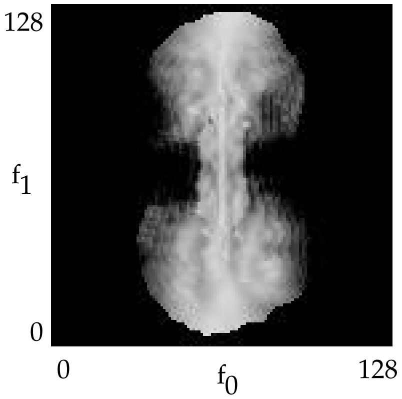
Slice through GGoF space at f_0=64
using an oriented normal GGoF function
Figure 5.41



Slice through GGoF space at f_1=64
using an oriented normal GGoF function
Figure 5.42



s value of local max thru s in GGoF
given an initial s of 6
using a full-rank oriented GGoF function
Figure 5.43

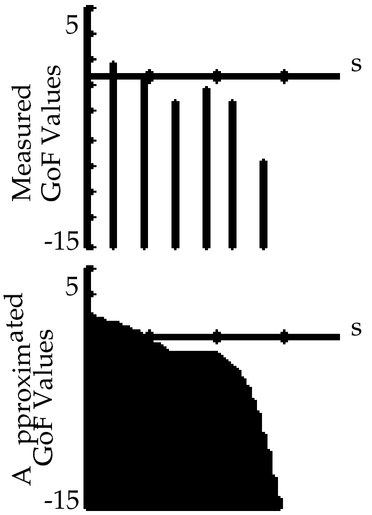


Value at local max thru s in GGoF
given an initial s of 6
using a full-rank oriented GGoF function
Figure 5.44

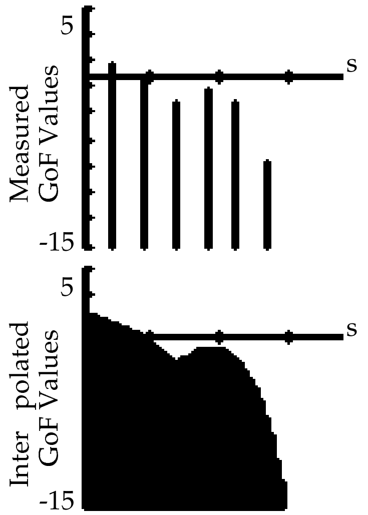
5.6.2. Interpolation versus Approximation

The use of approximating B-splines to provide GGoF space values has been questioned by several sources. Recent work by mathematicians at UNC has indicated that such approximations can adversely affect the number and location of singularities in GGoF space. Work performed for this dissertation has revealed the occurrence of such events. As a result, several alternative approximating and interpolating methods were investigated including higher order (5th) approximating B-splines, lower order (cubic) approximating B-splines, interpolating splines, Gaussian smoothing functions, interpolating quadratic polynomials, and methods based on Taylor series expansion.

Consider Figure 5.45. The upper plot depicts the measured GGoF values at f_0=65 and f_1=93 for s=4, 6, 8, 10, 12, and 14 for the training data in Fig. 5.15. This feature space point is one feature space unit away from an ideal core point for this data (f_0=64, f_1=93). In the lower plot, a quadratic B-spline approximation to the data is depicted for s=2.05, 2.1, 2.15, ... 14.95. Ideally, a local maximum exists at s=10. Such a maximum is clearly present in the measured values. The maximum is still present in the spline approximated values, but it is of significantly reduced relative magnitude. Compare these results with Figure 5.46. The lower plot depicts an "averaged Taylor series interpolation" of the data. The expected maxima is clearly visible. The reduction of the magnitude of local maxima is even more pronounced at other points in feature space when spline approximation instead of averaged Taylor series interpolation.



Measured and spline approximated GGoF values thru s at f=(65,93)
Figure 5.45



Measured and Interpolated GGoF values thru s at f=(65,93)
Figure 5.46

The equations of the "averaged Taylor series interpolation" employed are given below.

$$\hat{F}_0(x+h) = F(x) + hF'(x) + \frac{h}{2}F''(x) \qquad [5.7]$$

$$\hat{F}_1((x+1)-h) = F(x+1) - hF'(x+1) + \frac{h}{2}F''(x+1) \qquad [5.8]$$

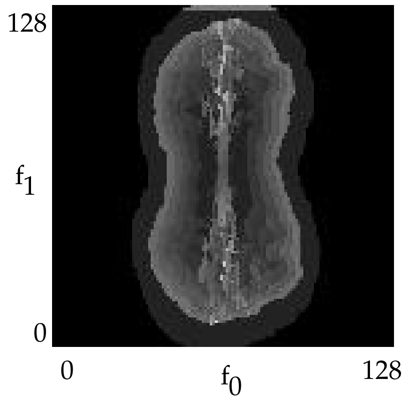
$$F(x+h) = (1-h)\hat{F}_0(x+h) + h\hat{F}_1((x+1)-h) \qquad [5.9]$$

where there derivatives are approximated using finite differences

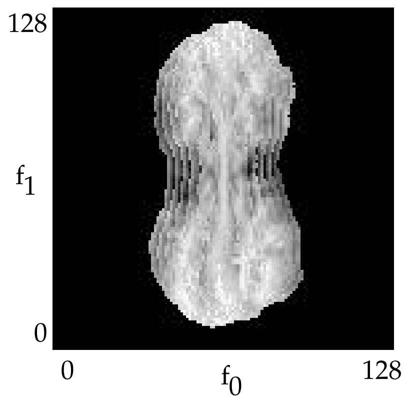
$$F'(x) = \frac{F(x+1) - F(x-1)}{2} \qquad [5.10]$$

$$F''(x) = \frac{F(x+1) - 2F(x) + F(x-1)}{2} \qquad [5.11]$$

Note that the cost in terms of the number of GGoF evaluations which must be performed per point evaluation is the same for approximating quadratic B-Splines and the averaged Taylor series interpolation technique. The effect of interpolation versus approximation is partially illustrated in Figures 5.47-5.48. These figures were generated under the same conditions as were Figures 5.45-5.46 except that the averaged Taylor series interpolation technique was used instead of the approximating quadratic B-splines.



s value of local max thru s in GGoF
given an initial s of 6
using an oriented normal GGoF function
Figure 5.47

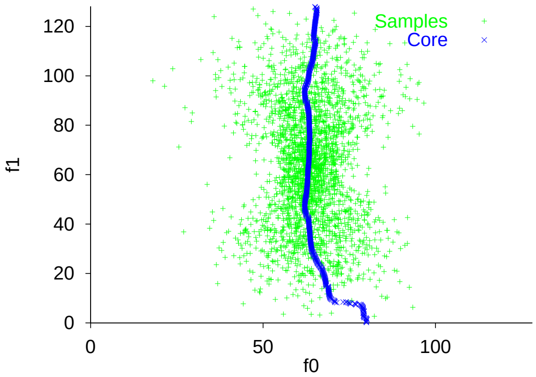


Value at local max thru s in GGoF
given an initial s of 6
using an oriented normal GGoF function
Figure 5.48

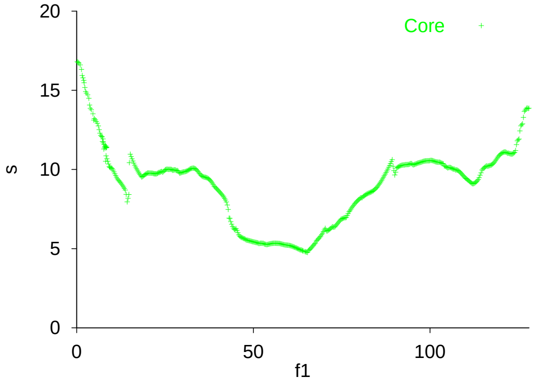
This dissertation provides the means by which these techniques and others can be compared, but it is beyond the scope and intent of this dissertation to perform such a comparison. For the remainder of this dissertation, approximating quadratic B-Splines will continued to be used. However, every integer scale, not just even-integer scales, will be explicitly evaluated. As discussed in Section 4.3, this is only possible when overlapped binning is used. Overlapped binning reduces the rapid change in GGoF possible when arbitrary consecutive scales are evaluated. This finer sampling of GGoF space should improve the correspondence between the measured and the approximated values.

5.7. GGoF Cores: An Example

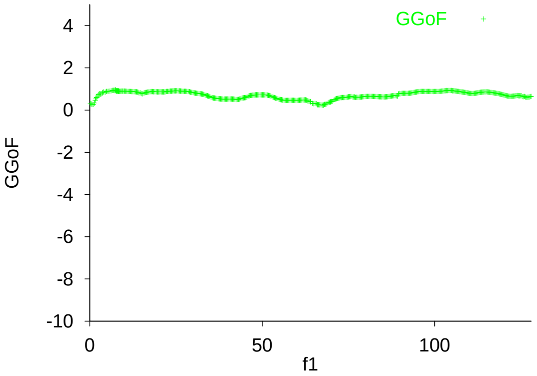
Two projections of the GGoF core of the training data are shown in Figures 5.49 and 5.50. This core was extracted using an automatically chosen stimulation point as discussed in Section 5.3.1.



Feature space projection of a core
Figure 5.49



Trace of core through s, f_1
Figure 5.50



Values of GGoF of core points
Figure 5.51

The trace of the ideal means appears to be well estimated by this core. The scale at the upper control Gaussian is slightly underestimated (estimated 10, ideal 14), but the rest of the correspondence is excellent.

Termination is a problem for GGoF cores as it is for medialness cores. Also, certain discontinuities are present in the GGoF core (Figure 5.50). Some preliminary work has been performed in both of these areas. The inclusion of stopping criterion based on the rate of change experienced between consecutive core points appears frequently to cause early core termination. Initial attempts at smoothing the core as a postprocessing step does appear to be beneficial and merits additional research.

5.8. GGoF Core to CGMM Conversion

Given N_t continua or "traces" $\mathbf{T}^{(j)}$ of Gaussians ϕ, a continuous mixture model provides a probability for a sample $\underline{x} \in \Re^N$ via

$$\mathbf{P}\left(\underline{x} \mid \Psi\right)= \underset{\{\Phi\}\in \mathbf{T}^{(j)}|j=1..N_t}{\mathbf{MAX}}\left(\mathbf{P}(\Phi)\mathbf{P}\left(\underline{x}|\Phi\right)\right) \qquad [5.12]$$

That is, this dissertation follows the missing data assumption of Dempster, Laird, and Rueben [Dempster, Laird et al. 1977] that each sample is in fact generated by just one of the infinite number of components, that the generating component is determined via maximum likelihood, and that the generating component provides the best estimate of the sample's probability. The focus of this dissertation is the definition of N_t continua of Gaussians ϕ $\mathbf{T}^{(j)}$ via core techniques and the estimation of their associated $\mathbf{P}(\phi)$ using traditional statistical methods.

This section shows that it is useful for the traces $\mathbf{T}^{(j)}$ to be GGoF cores. That is, for each GGoF core point ϕ the *a priori* probability $\mathbf{P}(\phi)$ can be estimated, and the points ϕ completely define a Gaussian classifier which can be used to provide a core point conditional sample probability $\mathbf{P}(\underline{x} \mid \phi)$.

5.8.1. Core Point *A Priori* Probability

The training samples from a particular class are used to maximize the parameters of the model of that class. Thus, a class's *a priori* probability is usually defined as the number of samples used to define that class divided by the total number of samples in the training set.

Each core point, $\phi=(\underline{\mu},s)$, is defined by a local collection of training samples; that is, all samples within a bounding box about $\underline{\mu}$ are binned to measure its GGoF value. Thus, a core point's *a priori* probability, $\mathbf{P}(\phi)$, is the number of samples in its local collection divided by the total number of samples in that class' training set.

5.8.2. Core Point Conditional Sample Probabilities

To provide a core point conditional sample probability at each core point $\phi=(\underline{\mu},s)$ a covariance matrix $\underline{\underline{\bullet}}(\underline{\mu},s)$ must be defined. The core point conditional sample probability is then

$$\mathbf{P}\left(\underline{x}\middle|\underline{\mu},s\right)=\frac{1}{(2\pi)^{N/2}\left|\underline{\underline{\Sigma}}(\underline{\mu},s)\right|^{1/2}}e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}\right)^t\underline{\underline{\Sigma}}(\underline{\mu},s)^{-1}\left(\underline{x}-\underline{\mu}\right)} \qquad [5.13]$$

A GGoF core can be parameterized by the explicit steps taken during the core traversal process. Any interpolation or approximation scheme can be used to determine intermediate core points. For this dissertation, no intermediate points were used since the steps made during the traversal process were small relative to the changes the probability surface that they represented.

The construction of a covariance matrix at each core point $\underline{\underline{\bullet}}(\underline{\mu},s)$ is dependent on the eigenvectors (as core normals and tangents) and eigenvalues of the projection matrix. To tie the core point covariance matrix definition to the core point extraction process, it was decided that if an eigenvector or eigenvalue was used to define the GGoF function, it should be used in the definition of the core point's covariance matrix.

Since adaptive-normal GGoF functions are being used to extract the GGoF cores of this dissertation, define

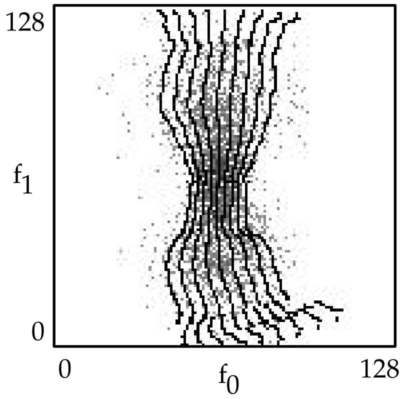
$\alpha^{(i)}(\underline{\underline{P}})$ = ascending ordered eigenvalues of $\underline{\underline{P}}$, i=1..Q
$\underline{v}^{(i)}(\underline{\underline{P}})$ = corresponding eigenvectors of $\underline{\underline{P}}$, i=1..Q
$\alpha^{(i)}(\underline{\underline{\bullet}}(\underline{\mu},s))$ = ascending ordered eigenvalues of $\underline{\underline{\bullet}}(\underline{\mu},s)$ i=1..N
$\underline{v}^{(i)}(\underline{\underline{\bullet}}(\underline{\mu},s))$ = corresponding eigenvectors of $\underline{\underline{\bullet}}(\underline{\mu},s)$, i=1..N

Then

$$\underline{v}^{(i)}\left(\underline{\underline{\Sigma}}(\underline{\mu},s)\right)=\begin{cases}\underline{v}^{(i)}\left(\underline{\underline{P}}\right) & i=1..Q\\ 0 & \text{otherwise}\end{cases} \qquad [5.14]$$

$$\alpha^{(i)}\left(\underline{\underline{\Sigma}}(\underline{\mu},s)\right)=\begin{cases}\dfrac{\alpha^{(i)}(\underline{\underline{P}})}{\max\limits_{j=1..Q}\left(\alpha^{(j)}(\underline{\underline{P}})\right)}s^{2} & i=1..Q\\ 0 & \text{otherwise}\end{cases} \qquad [5.15]$$

Using Equation 5.13 and this definition of the core point's covariance matrix, the isoprobability curves of the core point conditional probability can be visualized in feature space, Figure 5.51. Specifically, these curves are formed by plotting the 0, ±0.5, ±1, ±1.5, and ±2 standard deviation points normal to each core point.



Isoprobability curves of the core point conditional probability function
Figure 5.51

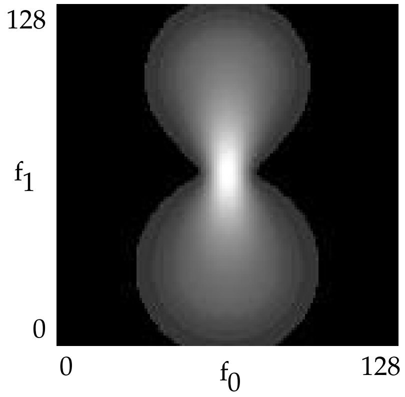
For adaptive normal GGoF functions, Q < N, and thus $\underline{\underline{\bullet}}(\underline{\mu},s)$ is of reduced rank. When Equation 5.13 is evaluated, any angled deviation from the core's normal decreases the core point conditional probability. This degradation can be controlled by incorporating the core tangent directions into $\underline{\underline{\bullet}}(\underline{\mu},s)$ and assigning them a standard deviation proportional to s.

5.9. Putting It All Together

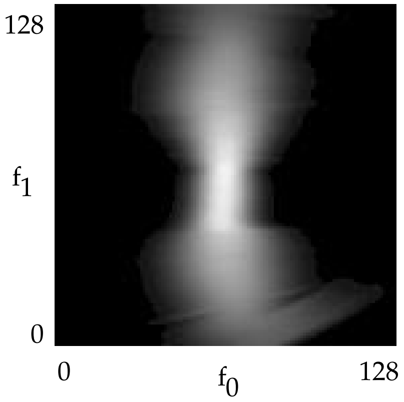
Figure 5.52 shows the probability density function of the population from which the training samples were generated.

By applying Equation 5.12 to every point in feature space using the GGoF core depicted in Figures 5.49 and 5.50 and allowing a standard deviation of 0.1s in the core tangent direction, Figure 5.53 is produced. Allowing a standard deviation of 1s in the core tangent direction increases the smoothness of the estimate (Figure 5.54) and therefore is used in Chapter 6.

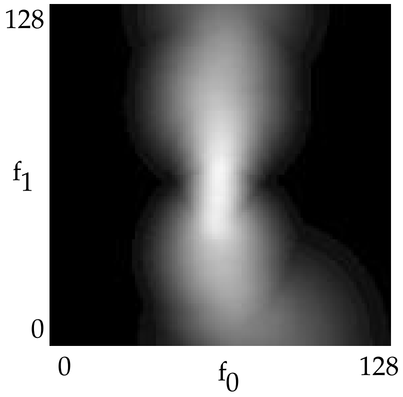
There subjectively appears to be excellent correspondence between the estimated and the actual density functions. Chapter 6 focuses on quantifying the accuracy and consistency of these estimates in terms of the accuracy and consistency of the classifiers they define.



Population's density function
Figure 5.52



Density function of the CGMM using tangent vector std. dev. of 0.1s
Figure 5.53



Density function of the CGMM using tangent vector std. dev. of 1s
Figure 5.54

5.10. Bibliography

Barnett, V. (1976). "The ordering of multivariate data." Journal of the Royal Statistical Society Series A **139**: 318-354.

Dempster, A., N. Laird, et al. (1977). "Maximum Likelihood for Incomplete Data via the EM Algorithm." Royal Statistical Society **1**(1)

Eberly, D. (1996). Ridges in Image and Data Analysis. Dordrecht, Kluwer Academic Publishers.

Eberly, D., R. B. Gardner, et al. (1994). "Ridges for Image Analysis." Journal of Mathematical Imaging and Vision **4**: 351-371.

Fritsch, D. S., S. M. Pizer, et al. (1994). Cores for Image Registration. Medical Imaging '94: Image Processing, SPIE.

Press, W. H., B. P. Flannery, et al. (1990). Numerical Recipes in C. Cambridge, Cambridge University Press.

Read, T. R. C. and N. A. C. Cressie (1988). Goodness-of-fit statistics for discrete multivariate data. New York, Springer-Verlag.

Chapter 6

BEHAVIOR OF GAUSSIAN GOODNESS-OF-FIT CORES

There is nothing more irritating than a good example.

- Mark Twain

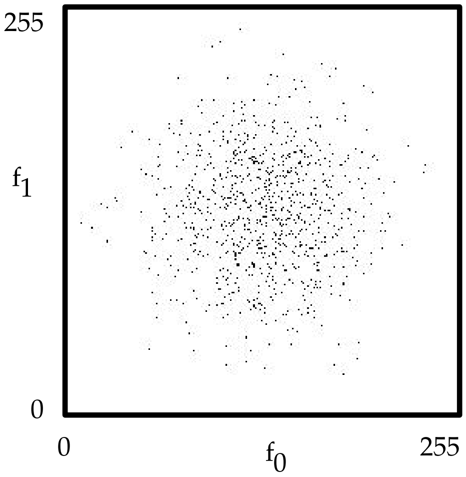
This chapter evaluates the accuracy and consistency of continuous Gaussian mixture modeling via Gaussian goodness-of-fit cores. Emphasis is placed on comparing CGMM via GGoF cores with K-means and FGMM. This chapter also demonstrates the ability of CGMMs to represent a trivariate extruded Gaussian distribution and classify gray and white matter in an inhomogeneous magnetic resonance image.

6.1. CGMM's Accuracy and Consistency

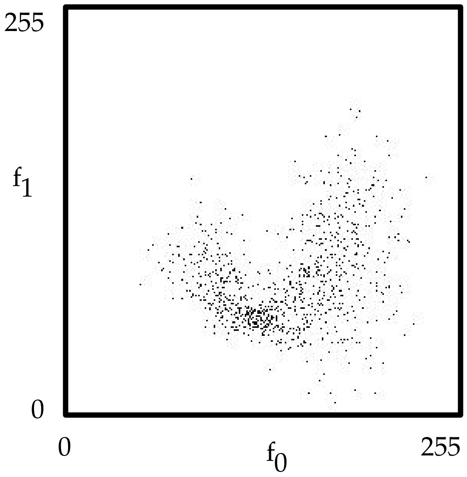
This section begins by detailing the operation of a CGMM classifier on a set of training and testing samples from the distributions of Problem 1 (Section 2.1.1.1). This analysis leads to the redefinition of the Problem 1 Monte Carlo study in consideration of the computational costs and distribution assumptions of CGMMs. The new study is used to quantify the accuracy and consistency of CGMMs, FGMMs, and K-means classifiers. Basic receiver operating characteristic (ROC) analysis is performed.

6.1.1. An Example CGMM Result

The following example is based on one run of the Monte Carlo simulation presented in Chapter 2. The 900 training samples from Class A used in the following example are shown in the scattergram in Figure 6.1. The 900 training samples from Class B are shown in Figure 6.2.

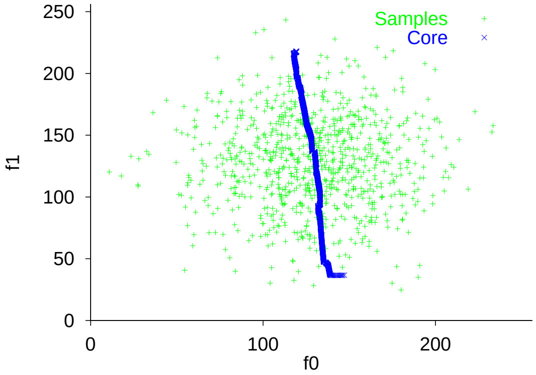


Class A's training data's scattergram
Figure 6.1

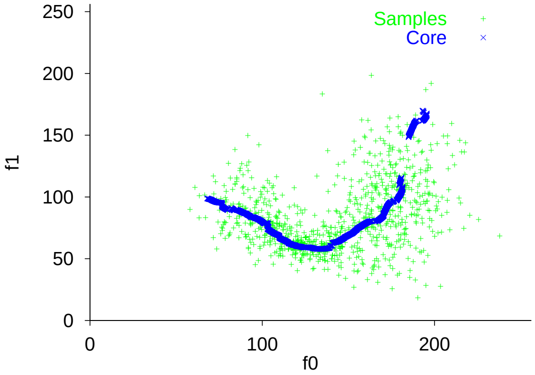


Class B's training data's scattergram
Figure 6.2

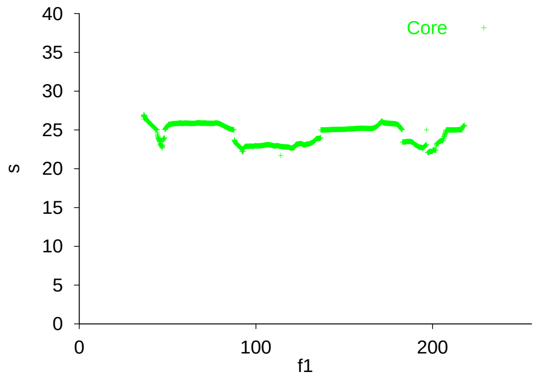
The algorithm presented in Section 5.3.1 automatically chose the feature space points (160.37,123.30) for Class A and (163.66, 80.08) for Class B for GGoF core stimulation. The chosen s_0 values were 26.25 and 17.94 respectively. The feature space projection of the extracted GGoF cores are shown in Figures 6.3 and 6.4. The effect of automated core recovery is illustrated by the missing section of the Class B core. These cores respectively span approximately 183 and 206 feature space elements. Their traces through scale are shown in Figures 6.5 and 6.6. Their collection of GGoF values are in Figures 6.7 and 6.8. The isoprobability curves of the core point conditional sample probabilities are shown in Figures 6.9 and 6.10. The estimated probability density functions are shown in Figures 6.11 and 6.12.



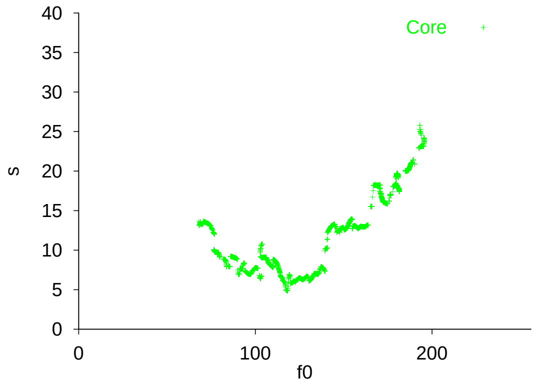
Feature space projection of the Class A core
Figure 6.3



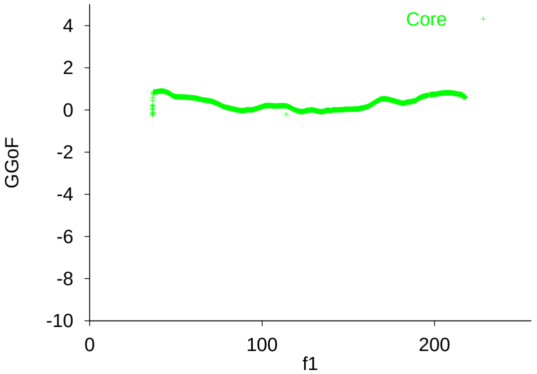
Feature space projection of the Class B core
Figure 6.4



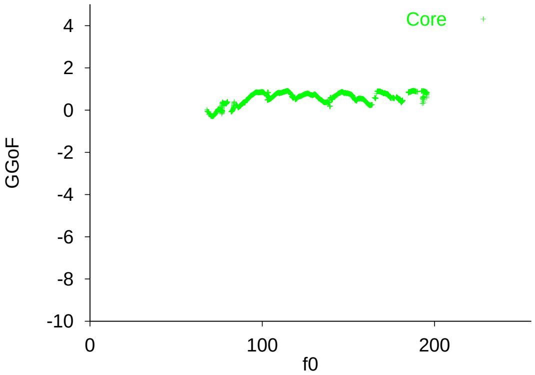
Track of Class A core through scale
Figure 6.7



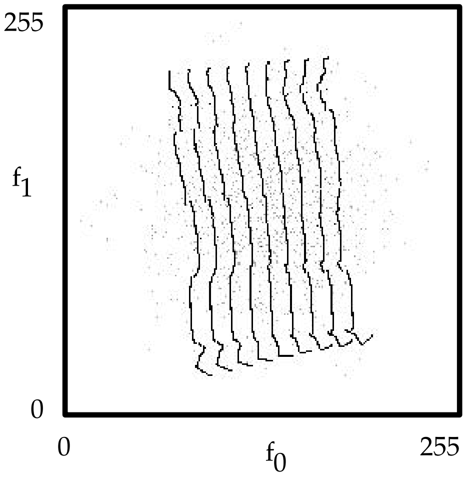
Track of Class B core through scale
Figure 6.8



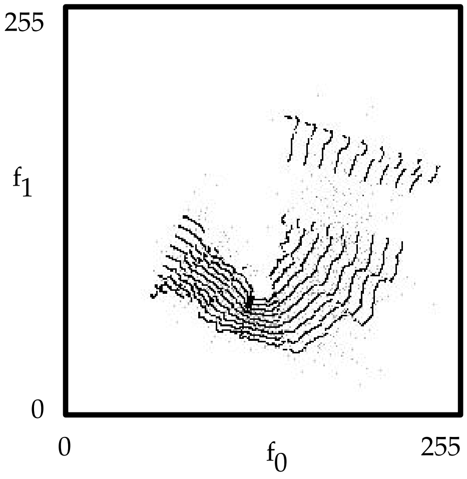
Track of Class A core through GGoF
Figure 6.11



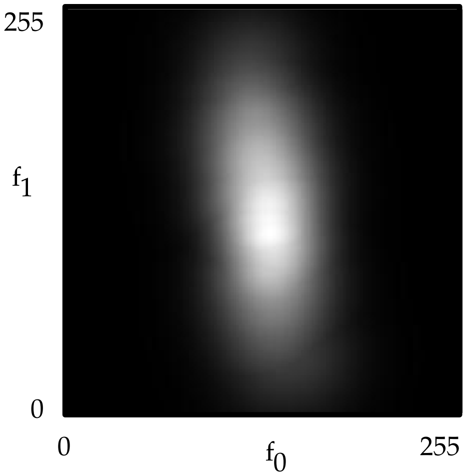
Track of Class B core through GGoF
Figure 6.12



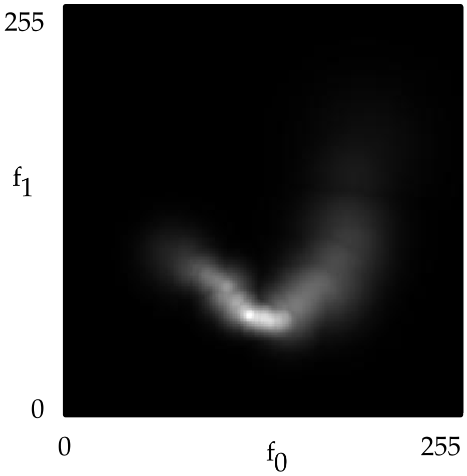
Isoprobability curve of the core point conditional sample probabilities for Class A
Figure 6.9



Isoprobability curves of the core point conditional sample probabilities for Class B
Figure 6.10

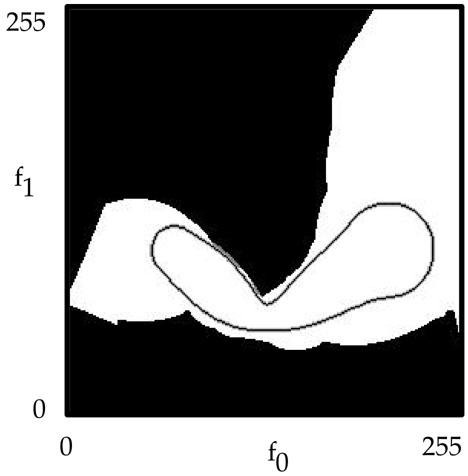


Estimated Density Function for Class A
Figure 6.13



Estimated Density function for Class B
Figure 6.14

Using the estimated probability density functions from these two classes, every point in feature space can be assigned a label, and a scattergram can be developed which reflects those labelings with differing shades of gray. Such a plot is given in Figure 6.15.

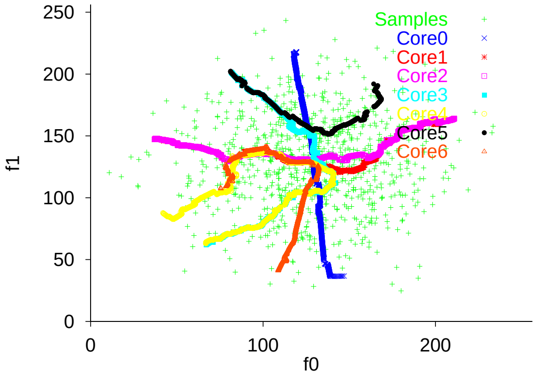


Labeling of feature space
using CGMM via 1 GGoF core
Figure 6.15

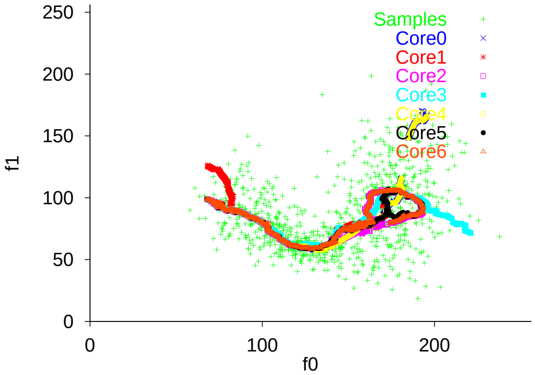
Apparently an accurate representation of the majority of the extent of the extruded Gaussian distribution can be generated. Both distribution representations, however, demonstrate poor GGoF core endstopping and occasionally poor estimates of variance.

Core endstopping is also a problem for medialness cores. For GGoF cores, however, the *a priori* probabilities of these overextended core points are lower so their negative effects are somewhat reduced.

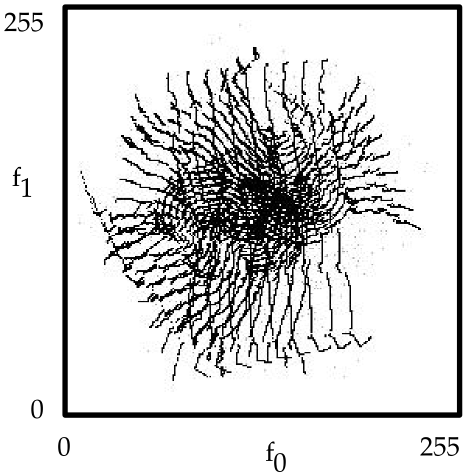
To improve the estimates of the local variance, multiple cores can be extracted. Figures 6.16-6.19 depict the feature space projection and the isoprobability curves of all 7 GGoF cores which can be extracted from the distribution when FGMM7 is used to provide the stimulation points.



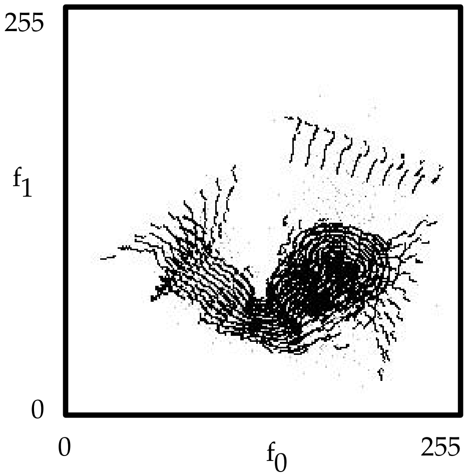
Trace of 7 cores of Class A's training data
Figure 6.16



Trace of 7 cores of Class B's training data
Figure 6.17



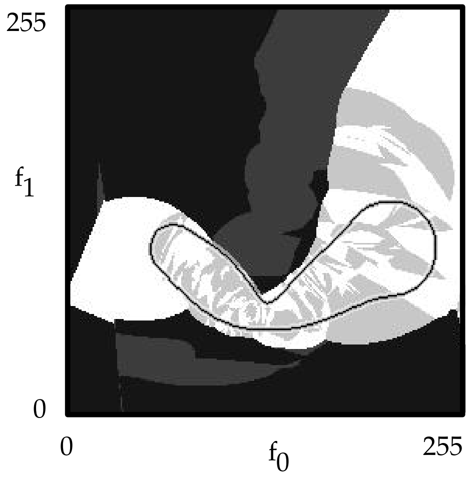
Class A's isoprobability cures
Figure 6.18



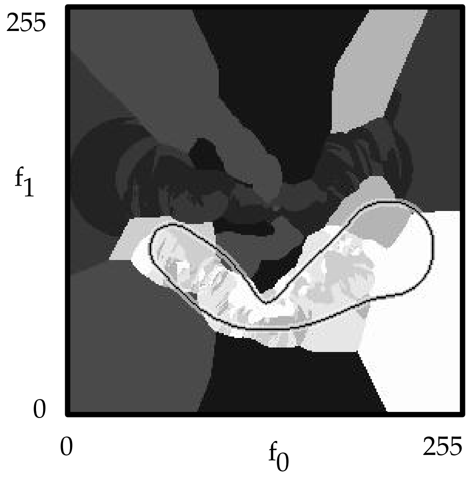
Class B's isoprobability curves
Figure 6.19

While the majority of the extent of these additional cores form duplicate representations, they are often able to fill in the gaps or provide better estimates of variance for portions of the extruded Gaussian distribution. Consider Class B; generally redundant representations are formed by the additional cores. Class B's generalized projective Gaussian nature, its match with the assumptions of CGMM, make the formation of redundant representations likely. The Class A distribution, however, is a nongeneric instance of a generalized projective Gaussian distribution. It is ideally represented by just a single Gaussian, i.e., a zero dimensional core. Using a 1D GGoF core to represent Class A overfits the data. As a results, there is less consistency in the multiple cores except at the center of feature space where the zero dimensional core would be likely to form.

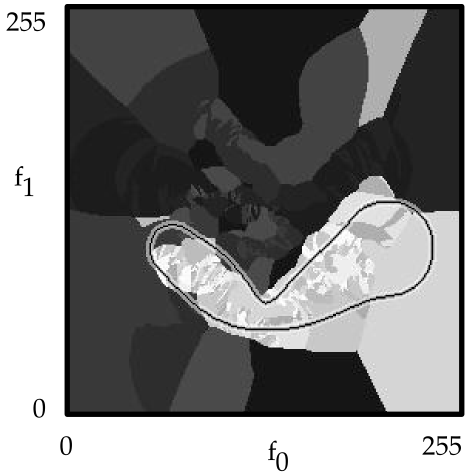
The labelings of feature space resulting from the use of 2, 4, and 7 cores per class are shown in Figures 6.20 to 6.22. Even when the assumptions of the CGMM do not match the distribution as with Class A, the use of additional cores seems to refine the representation of the distribution; they do not appear to confound the representations as did the use of additional components in FGMMs.



Labelings of feature space using
2 cores per class
Figure 6.20



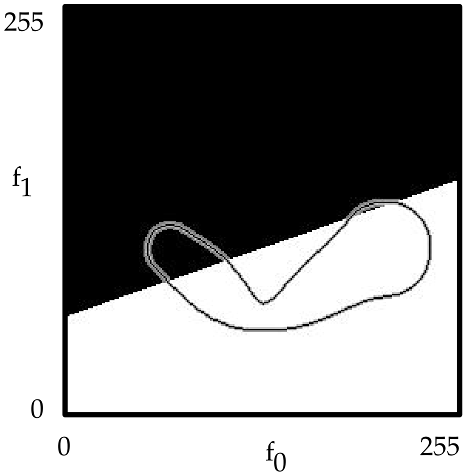
Labelings of feature space using
4 cores per class
Figure 6.21



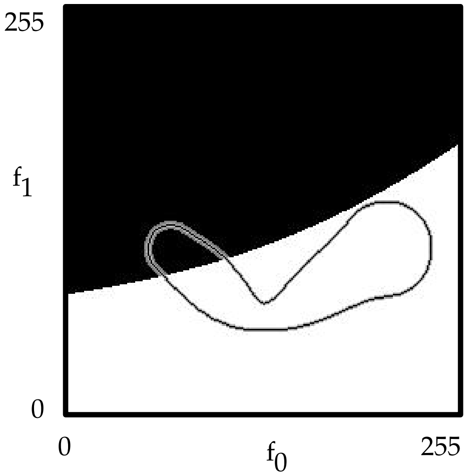
Labelings of feature space using
7 cores per class
Figure 6.22

For comparison, Figures 6.23-6.30 show the labeling of feature space provided by K-means and FGMM using 1, 2, 4, and 7 components and the training data in Figures 6.1 and 6.2.

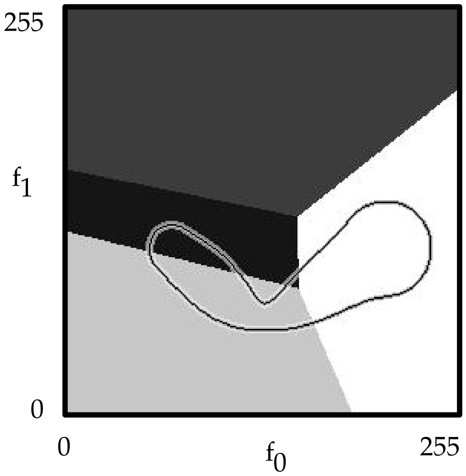
As discussed in Sections 2.3.6 and 2.3.7, the dependence of these classifiers on the user's specification of an appropriate K value is clear as is the presence of local maxima. For example, one of FGMM7's Class B components represents a sliver through feature space. That component is being poorly utilized, and its use does not correspond with the shape of the underlying distribution.



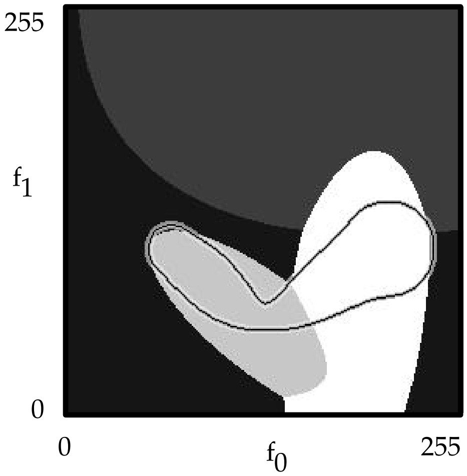
Labeling of feature space
using K-means with K=1
Figure 6.23



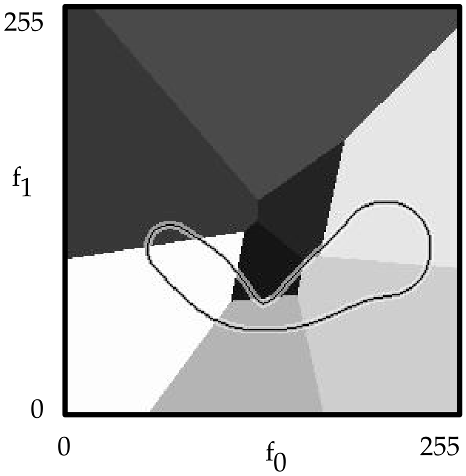
Labeling of feature space
using FGMM with K=1
Figure 6.24



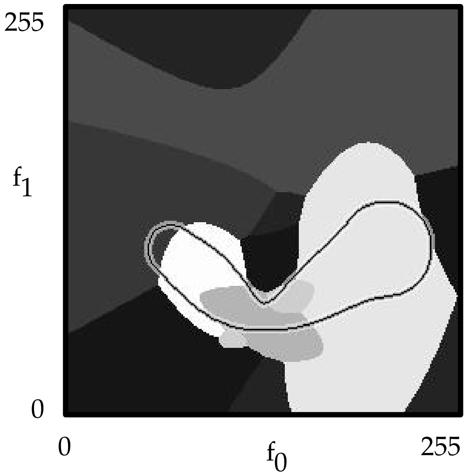
Labeling of feature space
using K-means with K=2
Figure 6.25



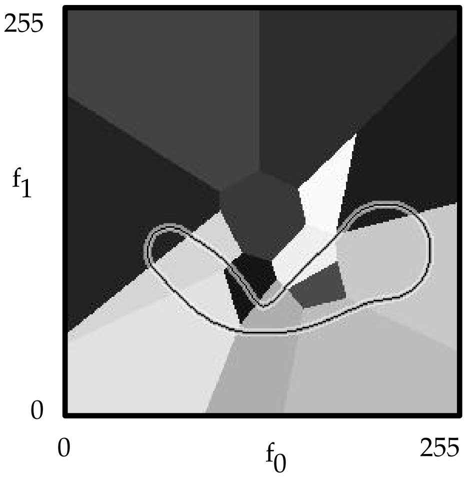
Labeling of feature space
using FGMM with K=2
Figure 6.26



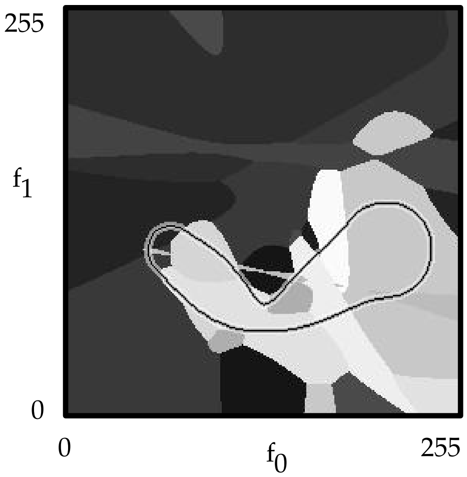
K-means with K=4
Figure 6.27



FGMM with K=4
Figure 6.28



Labeling of feature space
using K-means with K=7
Figure 6.29

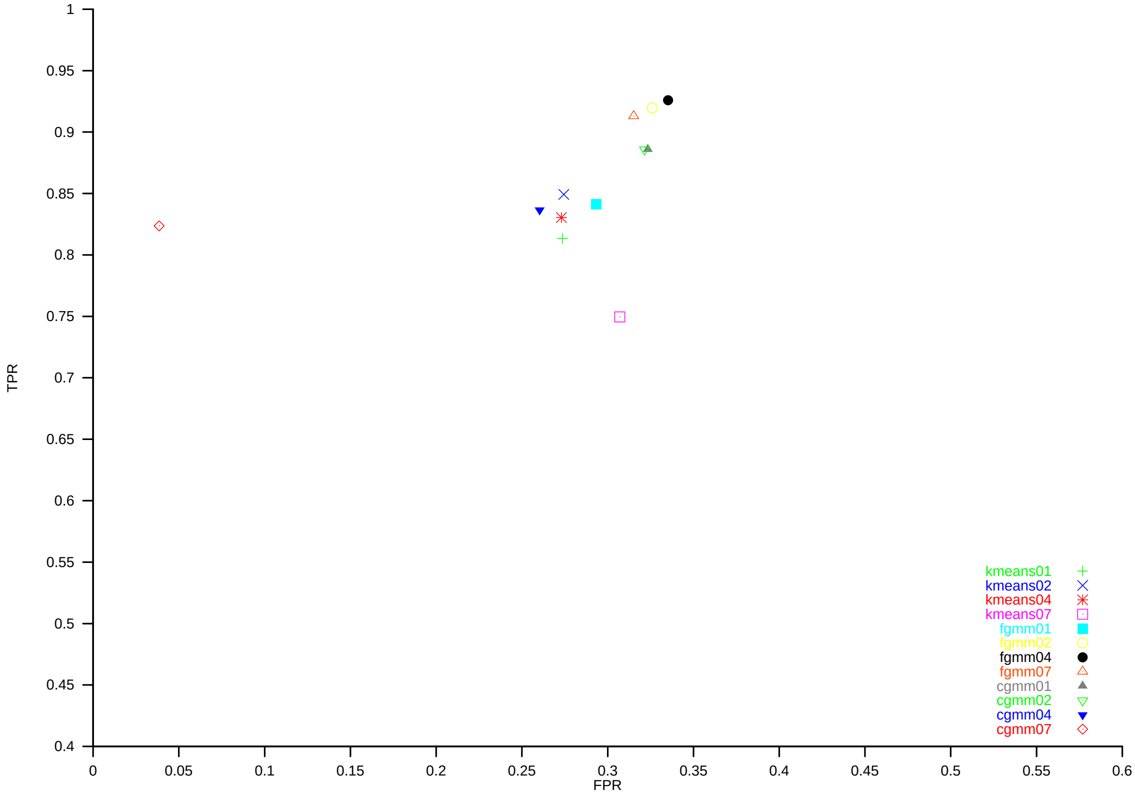


Labeling of feature space
using FGMM with K=7
Figure 6.30

The performance of CGMM via GGoF cores, K-means, and FGMM can be quantified as was done in Chapter 2. Given 2700 testing samples from each class, the FPRs and TPRs in Table 6.1 result. These values are summarized in Figure 6.31.

	K	FPR	TPR
CGMM	**1**	0.3233	0.8859
	2	0.3215	0.8859
	4	0.2604	0.8367
	7	0.0385	0.8237
K-Means	**1**	0.2737	0.8133
	2	0.2744	0.8492
	4	0.2730	0.8304
	7	0.3070	0.7496
FGMM	**1**	0.2933	0.8415
	2	0.3259	0.9196
	4	0.3315	0.9259
	7	0.3152	0.9130

TPR/FPR rates for the training data in Figures 6.1 and 6.2
Table 6.1

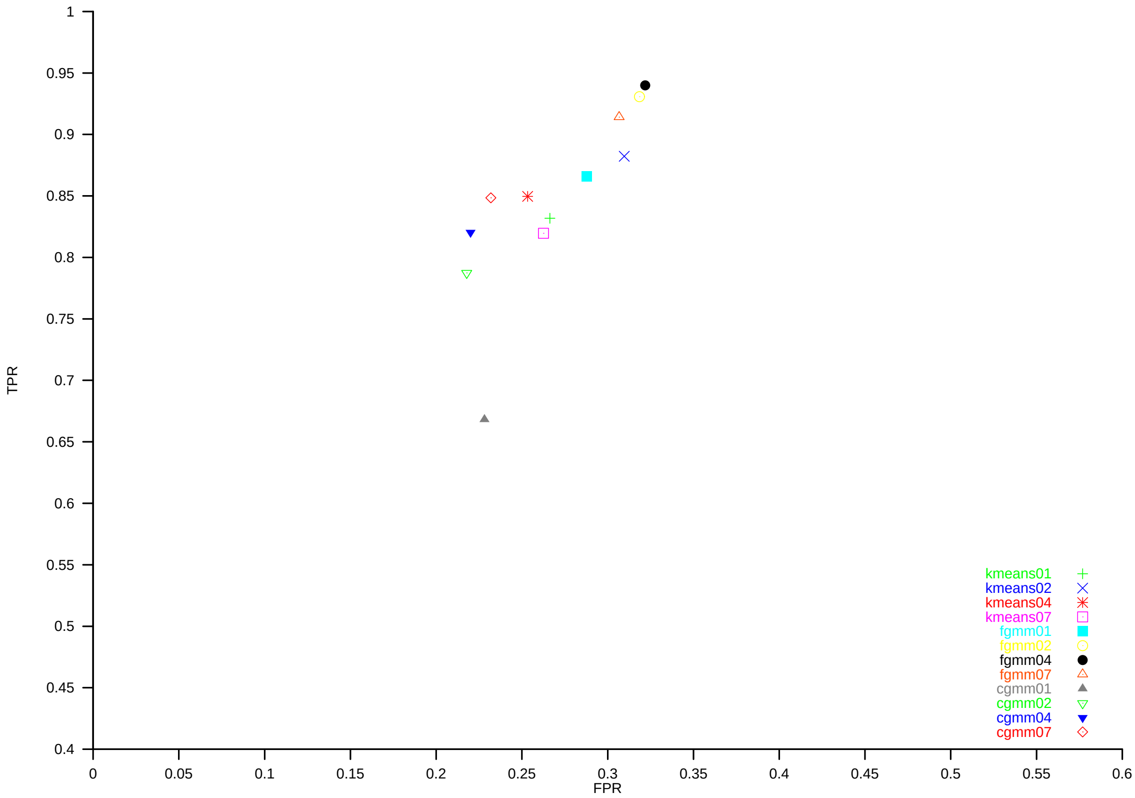


Plot of FPR/TPR for the training data in Figures 6.1 and 6.2
Figure 6.31

Compared to FGMM7, CGMM using 7 cores offers an significant decrease in the false-positive rate with only a small decrease in the TPR! To determine whether these results were anomalous, the experiment was repeated using a different random seed to generate the training and testing data. The FPRs and TPRs for the second run are summarized in Table 6.2 and Figure 6.31. As with the first set of data, CGMM using 7 cores produces a lower FPR while undergoing only a slight reduction in TPR. The differences in the performance of the various classifiers, however, are much less drastic than in the first run.

	K	FPR	TPR
CGMM	**1**	0.2281	0.6681
	2	0.2178	0.7874
	4	0.2200	0.8204
	7	0.2318	0.8485
K-Means	**1**	0.2663	0.8318
	2	0.3096	0.8822
	4	0.2533	0.8496
	7	0.2626	0.8196
FGMM	**1**	0.2878	0.8659
	2	0.3185	0.9307
	4	0.3218	0.9400
	7	0.3067	0.9141

TPR/FPR rates for run 2
Table 6.2



Plot of FPR/TPR for CGMM and all FGMM classifiers from Table 6.2
Figure 6.32

While no general conclusions can be drawn from these two runs, the results are extremely encouraging. CGMM7 provides the lowest FPR values for competitive TPR values. There is a ordered progression in the FPR/TPR values of CGMM as the number of cores used is increased.

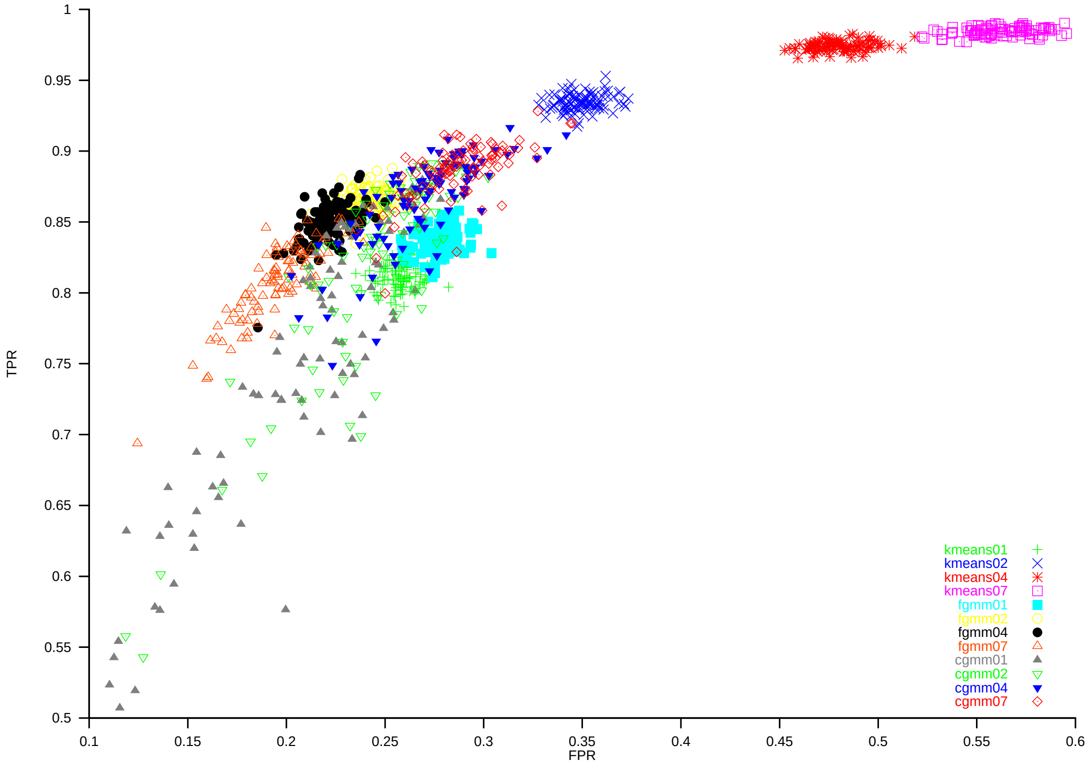
As revealed in the Monte Carlo experiments of Chapter 2, for K means and FGMM, the use of additional components only confounds the representation of these distributions. The ordered progression of CGMM's performance also suggests that it may perform more consistently than FGMM or K-means.

A Monte Carlo simulation is needed to better understand the behavior of CGMM via GGoF cores. The work in Chapter 2 already demonstrated FGMM's inconsistent performance on this problem, Section 2.3. Fewer conclusions can be drawn if CGMM also provides inconsistent performance on this problem. Additionally, the extraction of multiple cores greatly affects the computational cost of developing and operating the CGMM. If 14 cores of approximately 1500 points each are used, the complete processing of 1800 training and 5400 testing samples requires approximately 2.5 hours on a dedicated HP 712/100. To repeat the Monte Carlo experiments of Chapter 2 would require approximately 2500 hours! As a result, the Monte Carlo experiment was redesigned to take these performance histories and computational requirements into consideration.

6.1.2 Monte Carlo Results

The goal of all of the experiments in this dissertation is to evaluate the accuracy and consistency with which various techniques can model an extruded Gaussian distribution. Since Class A is a nongeneric extruded Gaussian, the Monte Carlo experiment used in Chapter 2 was redefined so as to limit the application of these technologies to the generation of an accurate representation of Class B. For every classifier in the new Monte Carlo simulation Class A is represented by a single Gaussian which exactly matches that population's parameters (Table 2.1). Additionally, the new Monte Carlo simulation involves only 100 runs. These changes reduce the time required to perform the simulation to 10 days.

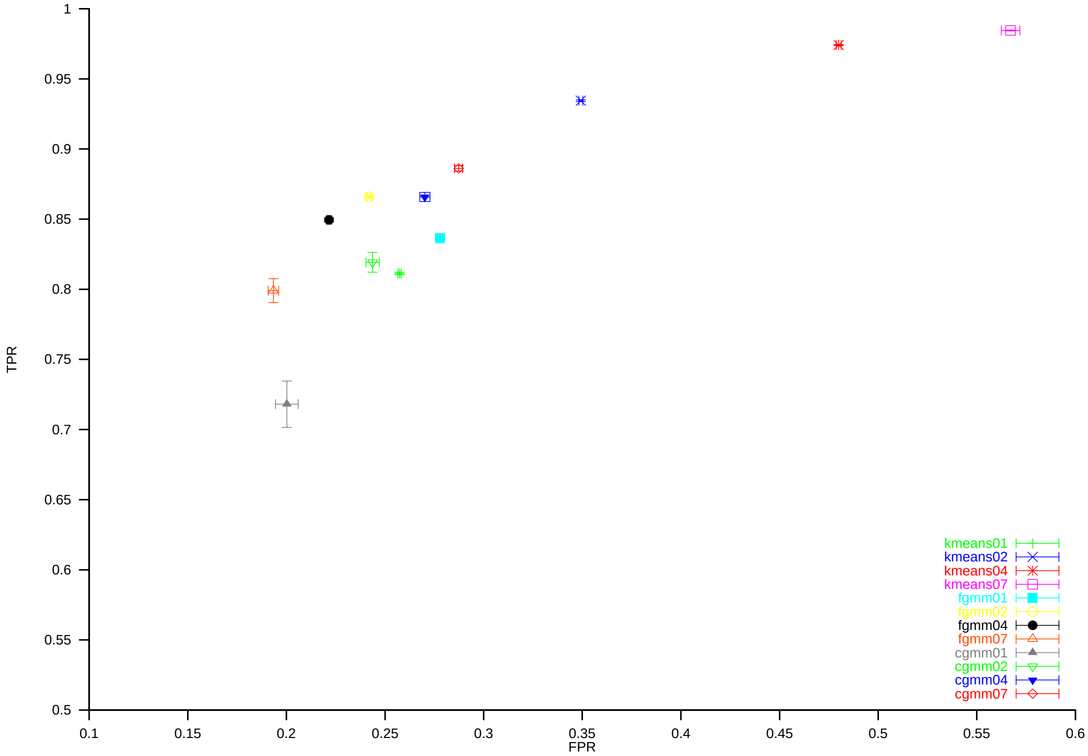
Each of the 100 Monte Carlo runs consisted of 900 Class B training samples, 2700 Class A testing samples, and 2700 Class B testing samples. The 100 FPR/TPR values which were recorded are show in Figure 6.33. The same performance measures that were used in Chapter 2 are used to compare the classifiers (see Section 2.2). The average TPRs, FPRs, and Monte Carlo one-sigma range shown in Table 6.3 result. The Monte Carlo statistics are depicted graphically in Figure 6.34.



The FPR/FPR values of the 100 Monte Carlo Runs
Figure 6.33

Method	FPR	TPR	FPR One Sigma	TPR One Sigma	Comb. One Sigma
cgmm01	0.2002	0.7181	0.0057576	0.0165489	0.0003563
cgmm02	0.2437	0.8192	0.0033732	0.0070245	0.0001220
cgmm04	0.2702	0.8658	0.0025880	0.0032410	0.0000544
cgmm07	0.2873	0.8862	0.0020565	0.0019929	0.0000308
kmeans1	0.2573	0.8112	0.0008511	0.0008505	0.0000072
kmeans2	0.3493	0.9345	0.0009904	0.0005903	0.0000057
kmeans4	0.4800	0.9741	0.0012722	0.0003461	0.0000043
kmeans7	0.5671	0.9845	0.0047031	0.0003357	0.0000136
fgmm1	0.2779	0.8364	0.0009231	0.0009339	0.0000084
fgmm2	0.2419	0.8660	0.0010374	0.0009371	0.0000093
fgmm4	0.2216	0.8495	0.0011087	0.0014111	0.0000129
fgmm7	0.1934	0.7990	0.0027022	0.0084882	0.0001117

Average TPR/FPR values and their Monte Carlo one-sigma ranges
Table 6.3



Plot of average TPR/FPR values and their Monte Carlo one-sigma ranges
Figure 6.34

As was the case for the Monte Carlo experiments in Chapter 2, the expected relations between the classifiers are upheld (see Section 2.3.8). Additionally,

1) Compared to their results for Problem 1 in Chapter 2, both K-means and FGMM demonstrate better accuracy and consistency.
2) Every method demonstrates an ordered progression in accuracy and consistency based on their hyperparameter. 100 runs are sufficient for Monte Carlo convergence for this simple problem.
3) CGMM1, CGMM2, and CGMM4 offer slightly poorer accuracy compared to FGMM2, FGMM4, and FGMM7. However, they distinctly offer higher levels of accuracy than any K-means.
4) CGMM4 and CGMM7 offer reasonable consistency.
5) CGMM7 offers very competitive accuracy and consistency.

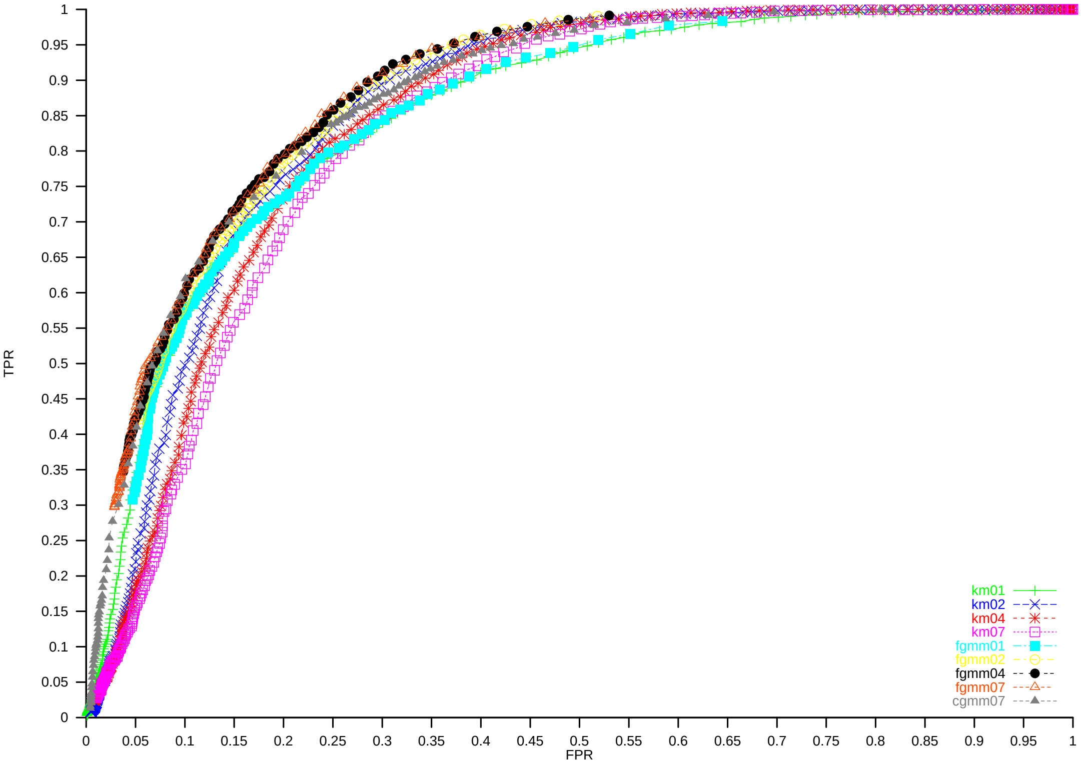
Thus, it can be concluded that CGMM via GGoF cores is a viable technique for representing extruded Gaussian distributions, but it is not definitive which classification technique is better. The next section attempts to determine which is better through ROC analysis.

6.1.3. ROC Analysis

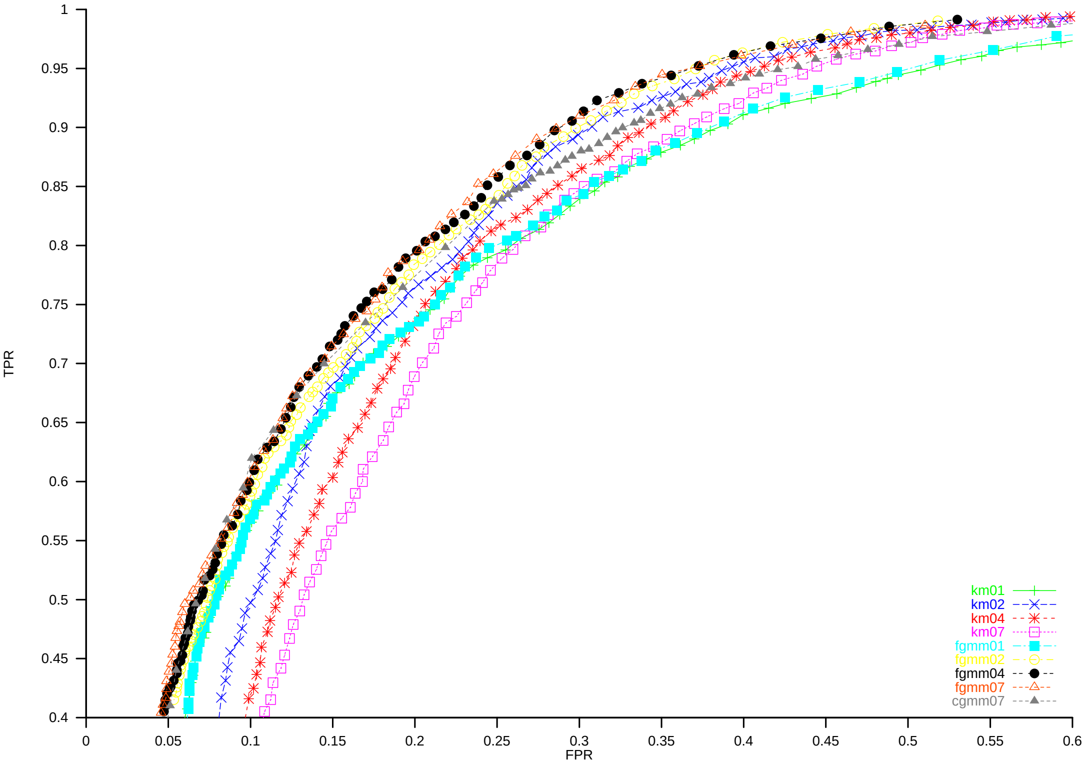
One run of this Monte Carlo experiment can be analyzed using ROC methods. [Egan 1975; MacMillan and Creelman 1991] By keeping the representations of the distributions fixed and changing the *a priori* probabilities, i.e., "observer biases", associated those representations, a continuum of FPR/TPR values can be generated. They form the ROC curves shown in Figure 6.35. An enlargement of the elbow of these curves is provided in Figure 6.36. Only CGMM7 is presented because of computational requirements.

There are some significant qualitative differences in these plots. Firstly, while the ROC curves for FGMM and CGMM are proper (nonincreasing slope), K-means produces an improper ROC curve. This is best illustrated in the sections of the ROC curves having near zero FPRs. Thus, changing the prior of Class A is not a proper variable for generating an ROC curve using K-means. Secondly, the ROC's associated with K-means are not ordered. That is, for K-means, given a change in K, the ROC plots may cross. It is difficult to tell, especially given just one example, if the FGMM ROC curves are ordered. Thirdly, the ROC curves are not symmetric about the negative diagonal through the graph. This suggests (correctly) that the two classes being considered do not have the same variance.

Three measures can be made to quantitatively compare these curves: the area under each of these curves can be calculated, the maximum probability of generating a correct answer can be calculated max-P(C)=Max(TPR+(1-FPR)), and a Neyman-Pearson observer comparison can be performed, i.e., compare TPR values given fixed FPR values [Egan 1975; MacMillan and Creelman 1991]. Table 6.4 summarizes these measures.



ROC curves from one run of the Monte Carlo Simulation
Figure 6.35



Enlargement of the elbows of the ROC curves
Figure 6.36

Method	**Area of ROC**	**Max-P(C)**	**TPR | FPR=0.1**	**TPR | FPR=0.15**	**TPR | FPR=0.2**
cgmm7	0.8752	1.5893	0.6160	0.7068	0.7741
kmeans1	0.8507	1.5478	0.5638	0.6707	0.7354
kmeans2	0.8604	1.5981	0.4974	0.6825	0.7644
kmeans4	0.8453	1.5659	0.4193	0.6033	0.7338
kmeans7	0.8306	1.5474	0.3566	0.5594	0.6898
fgmm1	0.8443	1.5530	0.5688	0.6704	0.7337
fgmm2	0.8665	1.6048	0.5889	0.6961	0.7844
fgmm4	0.8765	1.6126	0.6019	0.7166	0.7945
fgmm7	0.8793	1.6159	0.6047	0.7155	0.7935

Measures based on ROC curve from one training run

Table 6.4

This single ROC example indicates that

In regard to the area under the curves

1) The area under the CGMM7 curve is comparable to that of FGMM4 and only slightly less than FGMM7.

2) Clearly CGMM7 beats every K-means attempted.

In regard to the max-P(C) measure

1) CGMM7 provided performance similar to FGMM2, but well below FGMM4 and FGMM7

2) K-means2 provided competitive performance. This perhaps indicates the limited significance this measure has when based on just a single experiment or perhaps it indicates the appropriateness of K=2.

In regard to the TPR values at various FPR values

1) CGMM7 provided the best TPR value for the smallest FPR value tested (FPR=0.1). This fact together with the extremely low FPRs measured in Section 6.1.1 suggests that CGMM may be able to provide superior TPR values given small FPRs.

2) CGMM2, FGMM2, FGMM4 and FGMM7 outperform every K-means classifier.

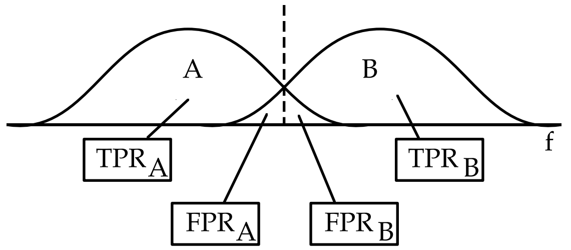
3) There is no ordered progression in performance with hyperparameter value for FGMM or K-means.

The next section makes use of the Monte Carlo average TPR and FPR (Section 2.2.2) to determine the expected ROC behavior of these classifiers.

* * *

The Monte Carlo analysis performed in Section 6.1.2 estimates the expected TPR and FPR values for the given problem for each classifier when maximum likelihood classification is performed. An ROC curve can be fit based on each classifier's FPR,TPR point if it is assumed that the distributions involved (Class A and Class B) are Gaussians having unit variance. While it is known that Class B is actually a generalized projective Gaussian and that the Class A and Class B distributions do not have equal variance, the unit variance Gaussian assumption does provide a first order approximation to the distributions.

By assuming unit variance Gaussians, the relative position of the means of the signal and noise distributions can be calculated from the average TPR and FPR values from each Monte Carlo run. Specifically, the TPR and FPR values define the portion of each distribution on either side of the decision bound (see Figure 6.37). These rates therefore define the distance from each distribution's mean to the decision bound.



TPR and FPR rates define the distance from the maximum likelihood decision bound to each mean
Figure 6.37

The parameter d′ in probit space analysis is equal to the spread of the estimated means [MacMillan and Creelman 1991]. Higher d′ values therefore indicate classifiers which are better able to distinguish between the classes. It can be assumed that classifiers with higher d′ values have better models of the distributions.

Using the d′ measures calculated from the Monte Carlo average TPR and FPR values from Section 6.1.2, the number of FGMM components that optimizes d′ can be determined. The relevant d′ values are given in Table 6.6. They indicate that FGMM using 2 components can be expected to provide the best model of the Class B distribution. The non-monotonic ordering of these values makes choosing an appropriate number of FGMM components difficult.

# of FGMM Components	**d′**
1	1.569
2	1.808
3	1.801
4	1.801
7	1.704
11	1.590

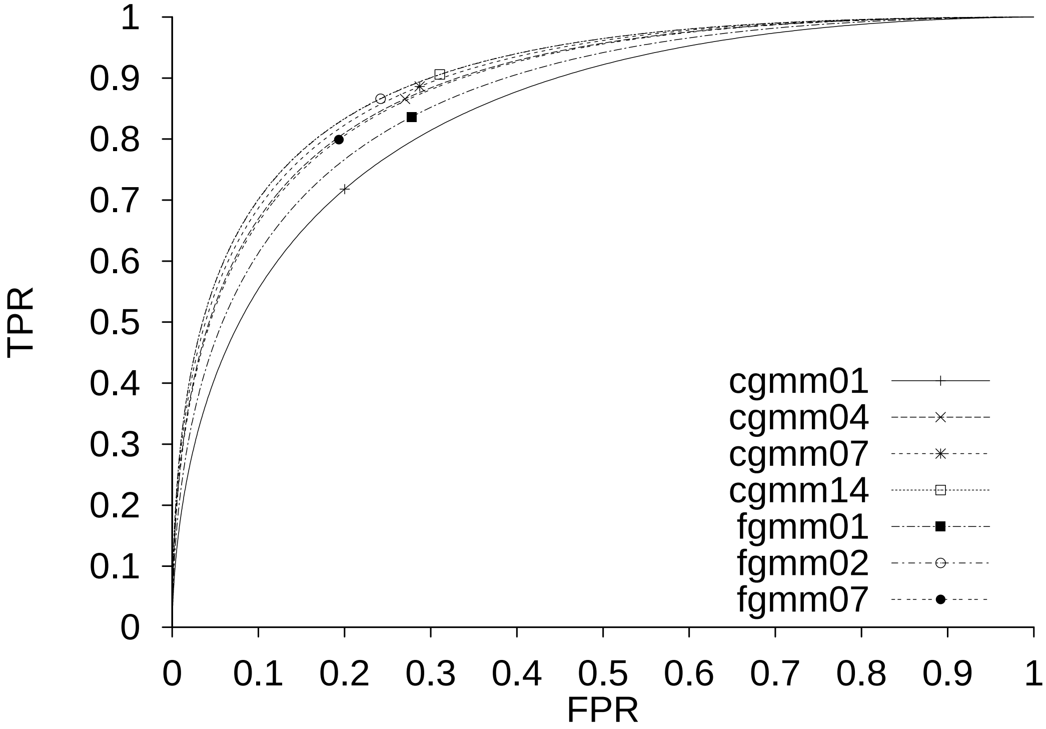
Spread of the means (d′) for differing numbers of FGMM components
Table 6.6

The expected spread of the means for CGMM can be studied with respect to the number of traces used to form the CGMM. Table 6.7 lists the relevant d′ values. These values clearly illustrate the asymptotic ordering of CGMM's performance as a function of the number of traces used. These values indicate that the expected performance of a CGMM using 14 or more components is better than that of the best FGMM model for the given problem.

# of traces in the CGMM	**d′**
1	1.418
2	1.607
4	1.719
7	1.768
14	1.810

Spread of the means (d′) for CGMMs with different numbers of traces
Table 6.7

Knowing the spread of the means d′ and assuming the distributions involved have unit variance, the complete ROC curve for each classifier can be calculated. Plots of these curves aid in the visualization of the differences in performance between FGMM and CGMM. Figure 6.38 provides such plots for select FGMM and CGMM configurations. CGMM14's ROC curve indicates that CGMM14 better models the GPG distribution Class B than the best FGMM model.



Estimated ROC curve passing through the Monte Carlo average performance values for CGMM and FGMM
Figure 6.38

6.1.4. Summary

All three analyses performed using generalized projective Gaussian distributions indicate that CGMM asymptotically provides better TPRs in low FPR situations compared to FGMM or K-means. Additionally, the consistency of CGMM is comparable to that of FGMM given problems for which FGMM actually provides consistent behavior.

Extracting multiple GGoF cores is shown to asymptotically refine the CGMM formed. This seems to be true even when nongeneric extruded Gaussian distributions are involved or when the use of additional FGMM components results in inconsistent FGMM behavior and unordered progressions in FGMM accuracy.

The next section demonstrates the application of CGMM via GGoF cores to a trivariate distribution. The subsequent section demonstrates the application of CGMM via GGoF cores to real-world data.

6.2. CGMM Representation of a Trivariate Distribution

This section presents a trivariate distribution and shows its CGMM representation. This increase in the dimensionality of feature space allows non-symmetric control Gaussians to be used to generate extruded Gaussian distributions having elliptical cross sections. Thus, the capabilities of the adaptive normal GGoF function are fully exploited.

The control Gaussians of this distribution are given in Table 6.8. Their use in the specification of a generalized projection Gaussian distribution is explained in Section 2.1.1.1. Because of the higher dimensionality of feature space, 9000 samples are used to represent the population. Figure 6.39 contains scattergrams formed by the projection of those samples onto f_2=0, f_0=0, and f_1=0.

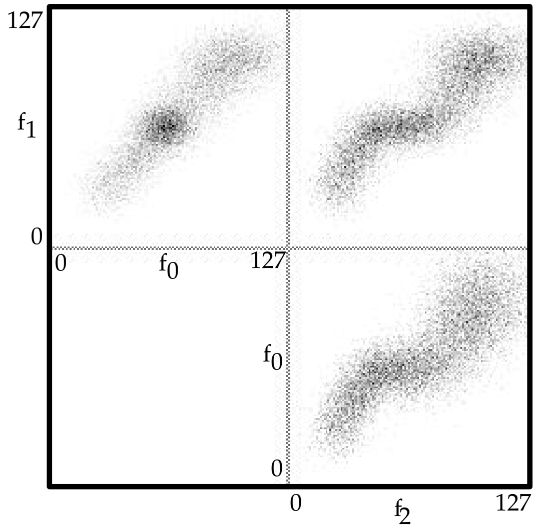
Gauss 0		**f0**	**f1**	**f2**
Mean		20.00	20.00	20.00
Covar	**f0**	64.00		
	f1		36.00	
	f2			144.00

Gauss 1		**f0**	**f1**	**f2**
Mean		74.00	74.00	44.00
Covar	**f0**	36.00		
	f1		20.25	
	f2			81.00

Gauss 2				
Mean		54.00	54.00	74.00
Covar	**f0**	36.00		
	f1		20.25	
	f2			81.00

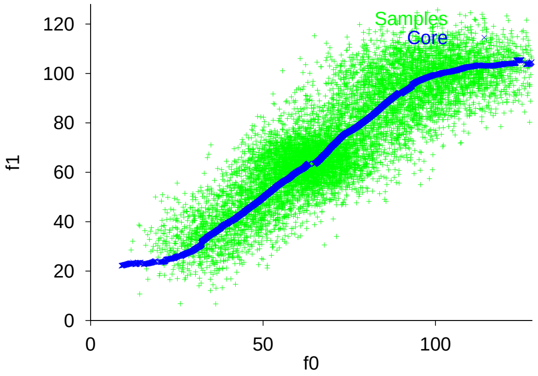
Gauss 3				
Mean		108.00	108.00	108.00
Covar	**f0**	64.00		
	f1		36.00	
	f2			144.00

The control Gaussians of the trivariate distribution
Table 6.8

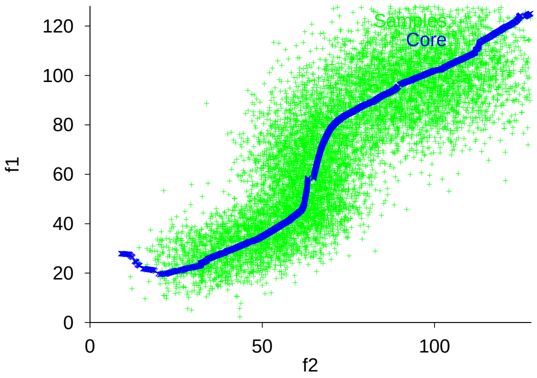


Scattergram of 9000 training samples
Figure 6.39

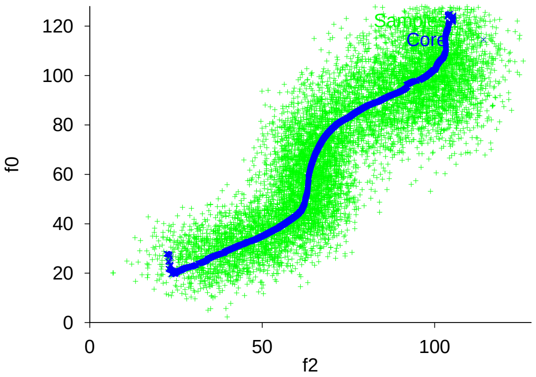
Using the algorithms defined in Section 5.2, a stimulation point at x=(96.20, 98.94, 66.04) and s_0=12.07 is automatically generated. A GGoF core spanning approximately 192 feature space volume elements is extracted. Its projections into feature space are shown in Figures 6.40-6.42.



Scatterplot with core overlaid
Figure 6.40

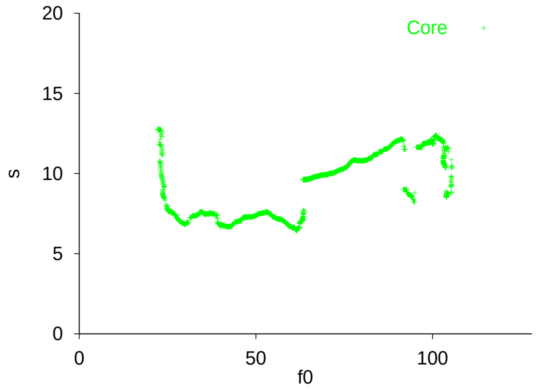


Scatterplot with core overlaid
Figure 6.41

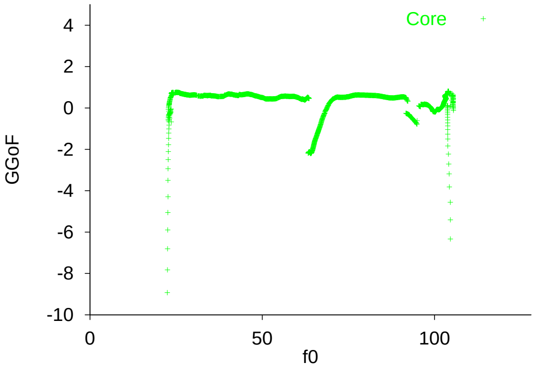


Scatterplot with core overlaid
Figure 6.42

The central skeleton of this distribution is well tracked by the GGoF core. The plot of the GGoF core through f_0, s is given Figure 6.43. The collection of GGoF values at the core points is shown in Figure 6.44.

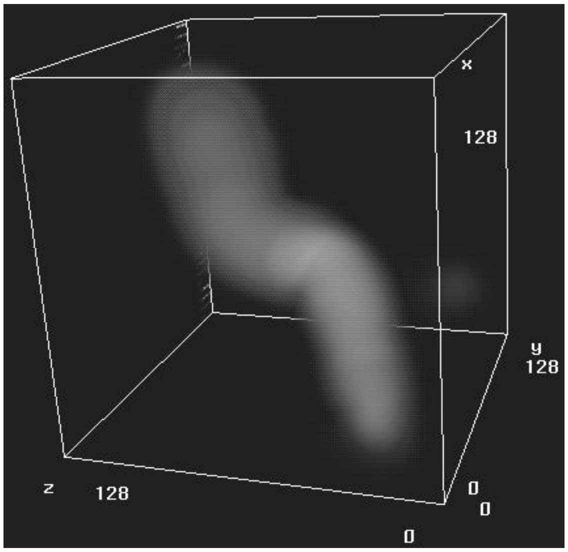


Track of Core through s
Figure 6.43



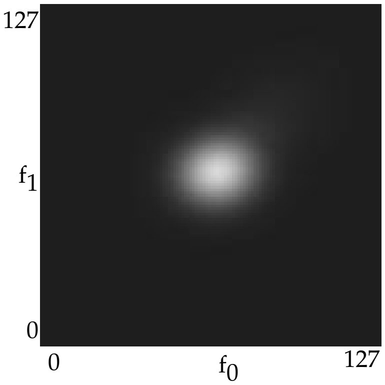
Collection of the Core's GGoF Values
Figure 6.44

The local scale is accurately estimated by the GGoF core throughout the majority of its extent. In the smaller variance regions of the distribution, the scale estimate goes astray, but not significantly. The rapid drop in the GGoF value at the overextended core points is obvious. A volume visualization of the CGMM is provided in Figure 6.45.



Estimated density function
Figure 6.45

By slicing through this distribution at f_2=64, the fact that the model maintains the elliptical cross section is demonstrated (Figure 6.46). The values of a GGoF core point in that region are given in Table 6.9. The expected ratio of the variances in the core normal directions (i.e., f_0 and f_1 for this core point) are present. The majority (~95%) of the core points evaluated demonstrated similarly accurate variance ratios.



Slice through the estimated density function reveals its elliptical shape
Figure 6.46

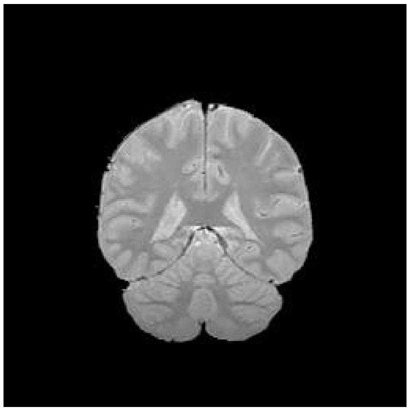
		f_0	f_1	f_2
Mean		65.738	64.223	63.673
Covar	f_0	43.728	2.160	8.095
	f_1	2.160	30.823	6.643
	f_2	8.095	6.643	77.639

Covariance matrix of an adaptive-normal GGoF core point captures the population's variance ratios
Table 6.9

In summary, the adaptive-normal GGoF function is able to trace an extruded elliptical Gaussian distribution in a three dimensional feature space. A one dimensional height ridge is tracked in the corresponding four dimensional GGoF space. A CGMM is defined. A probability density function is estimated. No user interaction is required. CGMM via GGoF cores operates successfully given distributions in high dimensional feature spaces.

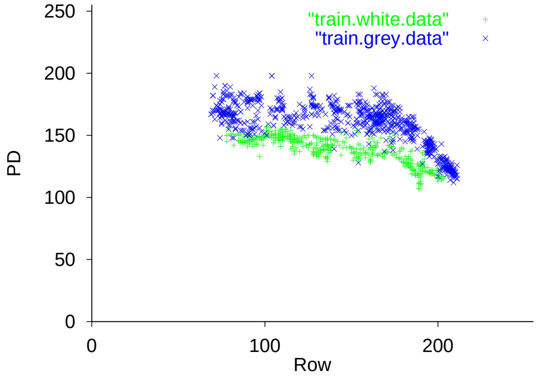
6.3. Classification of Tissues in Inhomogeneous Magnetic Resonance Images

This section demonstrates the efficacy of CGMM using GGoF cores given "real-world" data. As previously stated, a variety of medical imaging modalities demonstrate inhomogeneous distributions. This section will explain that MR is one such modality, and show that CGMM via GGoF cores can generate accurate representations of the tissue classes present in its data.



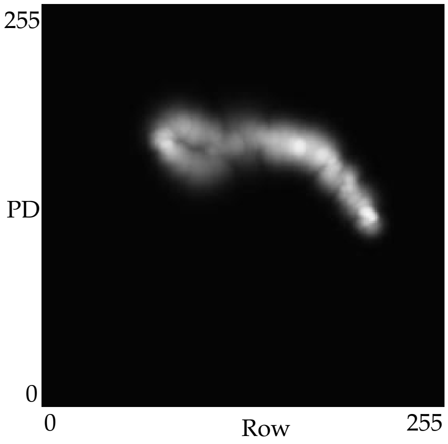
An inhomogeneous Proton Density MR Image
Figure 6.47

Consider the proton density (PD) MR image in Figure 6.47. A total of 984 white matter samples and 788 gray matter samples were collected from it to form a training set. A scatterplot of row, PD-value for the gray and white matter training sets is given in Figure 6.48. It clearly shows the overlapping of these distributions and the effect of the inhomogeneity. It is the row, PD-value information which is provided to each classification systems, i.e., f_0 = row and f_1 = PD-value.

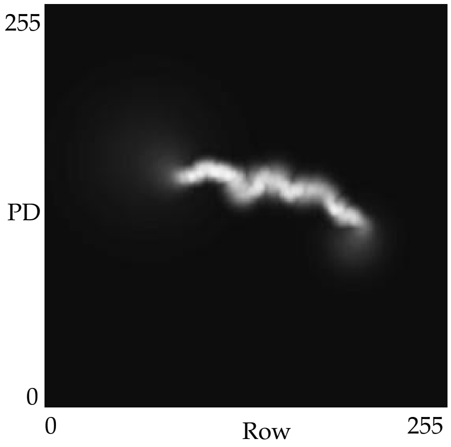


Scatterplot of the training data depicts the inhomogeneity
Figure 6.48

A single GGoF core is automatically extracted from each of the training sets. The estimated probability density functions of the corresponding CGMMs are given in Figures 6.49-6.50

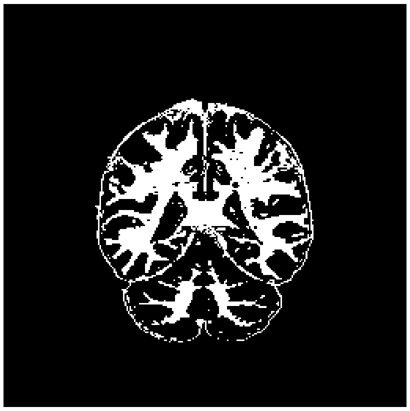


Estimate of gray matter density function
Figure 6.49

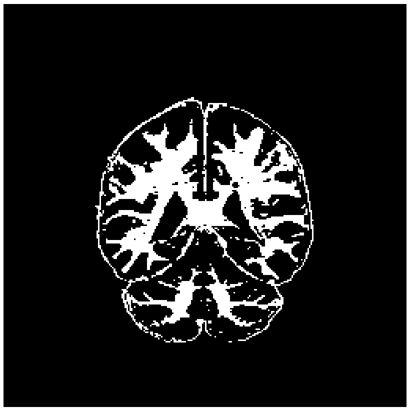


Estimate of white matter density function
Figure 6.50

Using the gray and white matter CGMMs, all of the pixels in the image, including those pixels which had previously been hand labeled and used for training, can be labeled (approximately 9% of the nearly 10,000 tissue pixels had were used for training). Even though other tissue types are present in the image, few non-white matter pixels should be assigned the white matter label since white matter generally has the darkest MR intensity. The collection of pixels assigned a white matter label are shown as an image mask in Figure 6.51. If FGMM is performed using 4 components per class, the white matter mask in Figure 6.52 results.

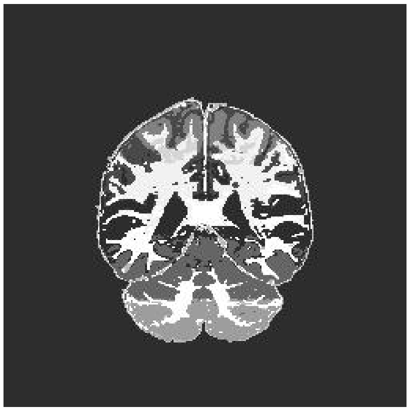


Mask of white matter from CGMM
Figure 6.51



Mask of white matter from FGMM4
Figure 6.52

These results indicate that CGMM is a viable alternative to FGMM for generalized projective Gaussian distributions given "real-world" data. The results from CGMM match those of FGMM however CGMM does not require the user to specify a K value, and CGMM does not suffer from poorly utilized components as FGMM does. Figure 6.53 contains a labeling of the image based on FGMM component membership. In the image, the significant detail is the allocation of four components (two per class) to the superior portion of the brain and the allocation of just two components (one per class) to the inferior portion where the effect of the inhomogeneity and the non-linearity of the distributions is the greatest. The components of the FGMM are being ineffectively used. As shown previously, the more extreme cases of poor FGMM component utilization can result in a reduction in the effective number of FGMM components. Such a "hidden" reduction confounds the user's task of selecting an appropriate number of components and increases the variability of the accuracy of the FGMMs formed for a population.



Labeling of PD image by FGMM component membership reveals poor component allocation
Figure 6.53

Chapter 7 summarizes and concludes this work.

6.4. Bibliography

Egan, J. P. (1975). Signal detection theory and ROC analysis. New York, Academic Press, Inc.

MacMillan, N. A. and C. D. Creelman (1991). Detection Theory: A User's Guide. Cambridge, Cambridge University Press.

Chapter 7

Conclusions, Summaries, and Future Work

As stated at the beginning of Chapter 1, most scientists encounter problems which involve statistical analysis. What populations are present in my data? How do these populations differ? Have I collected enough data? From which population did this sample originate? In statistical analysis, samples are used to form models of their source populations. This dissertation introduced a novel method for consistently and accurately forming models of data distributions. Questions such as those above are answered using measurements derived from such models.

Specifically, this dissertation presented a novel mechanism, Gaussian goodness-of-fit cores, for creating continuous Gaussian mixture models of generalized projective Gaussian distributions. In Monte Carlo studies against competing techniques, i.e., K-means and finite Gaussian mixture modeling, the proposed method is more automated, more accurate, and as consistent when representing generalized projective Gaussian distributions.

Generalized projective Gaussian distributions arise in a wide range of application areas. For example, although the intensity distribution of tissues within small regions of a magnetic resonance image (MRI) are Gaussian, smooth transforms must be applied to the parameters μ and • of these Gaussians to represent the same tissues in neighboring regions of that MRI. Therefore, to represent a tissue's intensities throughout an MRI, a generalized projective Gaussian must be modeled. Other researchers have shown that in controlled situations Gaussians can be used to represent features of a person's speech or handwriting, but smooth transforms must be applied to the parameters of those Gaussians to represent the same features when the data was acquired in different situations. That is, to represent features of speech and handwriting from multiple people in multiple situations, extruded Gaussians must be defined.

Finite Gaussian mixture models (FGMMs) are often used to model generalized projective Gaussian distributions. FGMMs use the weighted linear combination of multiple Gaussian component distributions to represent complex distributions. FGMMs are usually developed via maximum likelihood expectation maximization (MLEM). The user must select an appropriate number of components to be used by each FGMM, and that selection process is aggravated by the fact that MLEM has multiple, non-optimal local maxima. These maxima produce different models given different samples from the same distribution, and these maxima can result in the poor utilization of one or more components in a FGMM. The different models will provide different levels of accuracy, and this inconsistent accuracy is attenuated by poor component utilization in which the effective number of components is actually lower than the user specified number of components. The prevalence of non-optimal MLEM local maxima is expected to rapidly increase as the number of features or the number of components is increased.

GGoF cores represent continuous Gaussian distributions by tracking the continuum of means of those distributions and estimating the local variance of the distribution normal to that track. CGMMs formed from GGoF cores do not require the specification of a hyperparameter, e.g., the number of components of a FGMM. Monte Carlo simulations demonstrate that the accuracy and consistency of a CGMM improve asymptotically as the number of GGoF cores used to define the CGMM is increased and that CGMM provide better accuracy and consistency than FGMM for generalized projective Gaussian distributions. The application of CGMMs to real data is shown via the classification of gray and white matter in an inhomogeneous MRI. CGMMs are expected to scale well as the number of features increases.

Summary of Specific Results and Contributions

The first set of tests used Monte Carlo analysis to quantify the accuracy and consistency of competing classification methods for generalized projective Gaussian distributions. The Monte Carlo simulations involved two related classification problems having generalized projective Gaussian distributions. The average true positive and false positive rates were used to quantify the accuracy of the methods. The Monte Carlo one-sigma ranges of those rates were used to quantify the consistency of the methods. Linear, Gaussian, K nearest neighbor, Parzen windowing, multilayered perceptrons, K-means, and finite Gaussian mixture modeling were tested. Numerous expected relations between the performance of these classifiers were upheld in the experiments. The consistency of FGMM, however, was shown to degrade as the number of its components was increased. It is surmised that local maxima of the FGMM's parameter selection process, MLEM, was the source of the degraded performance. Also, the visualization of the labels of feature space provided by FGMM often illustrated the poor utilization of its components. A result of the poor utilization is a reduction in the effective number of components being used by the FGMM. This confounds the user's task of selecting an appropriate number of components.

With the goal of developing an accurate and consistent CGMMs, the accuracy and consistency of each of the key technologies of the CGMM development process, i.e., medialness cores and goodness-of-fit functions, were analyzed.

The accuracy and consistency of medialness cores have been detailed in other publications. In general, medialness cores do not vary significantly given a wide variety of image and object noise. As a result, medialness cores have been successfully used for registration as well as object recognition. The consistency and accuracy of medialness cores comes from their use of medialness functions to integrate information transverse to the core. Medialness functions, however, do not consider absolute image intensities. When applying core techniques to the representation of distributions, intensity corresponds to sample density and therefore is critical. Goodness-of-fit functions were investigated as an alternative technology for core generation.

The accuracy and consistency of a variety of goodness-of-fit functions were quantified using Monte Carlo simulations. In particular, this work sought to identify the accuracy and consistency with which the parameters producing a local maxima of a Gaussian goodness-of-fit function corresponded to a sampled Gaussian's actual parameters. A variety of sample set sizes, GGoF functions, and binning techniques were considered. The Monte Carlo simulations determined that the binning technique used to present the data to the GGoF function had the most influence on the accuracy and consistency of the GGoF's maxima. The log likelihood ratio function using overlapped-equiprobable binning provided the most accurate and consistent GGoF maxima.

Using this GGoF function and combining the core extraction techniques of Fritsch, Yoo, and Eberly, a core traversal technique, tuned for generating CGMMs, was defined. In particular, it was demonstrated that the eigenvectors of the local data's covariance matrix can be used to approximate the tangents and normals of the core; derivative information is therefore not needed to traverse the core. The associated eigenvalues can also be used to specify the expected variance in each of the core normal directions. Using this information, a one dimensional GGoF core can accurately model a continuous Gaussian distribution having an elliptical cross-section. Additionally, GGoF core termination based solely on its GGoF values was demonstrated to be effective.

The resulting GGoF cores provide core point conditional sample probabilities. Core point *a priori* probabilities are provided by estimates of the local training sample density. Together these constructs define a CGMM.

The accuracy and consistency of CGMM was demonstrated using four experiments.

1) The TPR/FPR of CGMM on one run of the Monte Carlo simulation used to evaluate the accuracy and consistency of the competing classifiers was measured.
2) A revised Monte Carlo simulation involving a simplified version of the original Monte Carlo study was performed.
3) A nonparametric ROC analysis of one run of the revised Monte Carlo experiment was conducted.
4) A parametric ROC analysis using the average TPR/FPR from the revised Monte Carlo experiment was made.

Every experiment supported the following conclusions

1) CGMM provides asymptotic better accuracy and consistency given additional GGoF cores. FGMM may not provide asymptotic accuracy or consistency given additional components.
2) CGMM provides equivalent consistency to FGMM for those problems for which FGMM provided asymptotically better consistency.
3) CGMM provided better TPRs for low FPRs than FGMM

Some of the experiments demonstrated that CGMMs can provide significantly better TPRs for low FPRs. The ROC analysis using the revised Monte Carlo averages indicated that CGMM may provide better accuracy with respect to the area under the ROC curve and the max probability of providing a correct answer, and CGMM may provide better TPRs for any FPR value.

Limitations of CGMMs via GGoF Cores

The use of CGMMs via GGoF cores, however, is not appropriate for ever statistical analysis problem. Four characteristics of such problems are:

1) If a more specific distribution assumption can be made, the corresponding distribution model will provide better accuracy and consistency.
2) If the parameter transformations which produced the continuum of Gaussians are well understood and easily measured, then a simple, more accurate, and more consistent Gaussian model may result from the application of the inverse of those transformations.
3) If model development time or the time to label a new sample are important, CGMM via GGoF cores may not be viable.
4) If only labeling of the data and neither its summarization nor the analysis of its model is important, a variety of non-parameteric methods such as Parzen Windowing are worthy of investigation.

Despite these constraints, however, numerous “hard” statistical pattern recognition problems such as those involving medical images, speech, and handwriting, remain to be solved by CGMM via GGoF cores.

Future Technical Advancements

Four areas stand out as avenues for future technical work: discretizing GGoF cores for FGMMs with better accuracy and consistency, investigating of unbiased GGoF functions, and transitioning lessons learned to medialness cores.

The models formed from the complete GGoF cores appear to be overly complex. This is apparent in the white matter CGMM which exhibited a looping behavior (Figure 6.x). The definition of FGMMs via GGoF cores requires the additional step of pruning the GGoF core points. This pruning will simplify the representation formed, can incorporate distribution smoothness constraints, and should result in further improvements in accuracy and consistency.

Improvements in accuracy, computational requirements, and memory requirements may come from the development of GGoF functions which provide unbiased evaluations at non-integer points in GGoF space (Section 6.X). Empirical distribution functions and Hermite polynomial based GGoF functions may provide solutions [REF]. If successful, the associated speedup and memory reduction would also facilitate the investigation of higher dimensional cores and feature spaces.

Most of the novel GGoF core techniques can be transferred to medialness cores. Two lessons learned are stated here. Firstly, the use of the local data's covariance matrix to define the core's normals and tangents is applicable to many segmentation tasks involving high contrast objects. For example, adaptive medialness functions using the local data's covariance matrix may aid in the extraction of vessels in magnetic resonance angiographic images of the brain or liver, or in the traversal of airways in x-ray computed topographic images of the lung. Secondly, the use of the eigenvalues of the local data's covariance matrix to specify an adaptive medialness function's shape warrants investigation. The projection of the local image data onto the core normals prior to medialness evaluation may also provide additional insensitivity to image and object noise. The projection technique may also facilitate the application of medialness cores to higher-dimensional images.

Going Beyond

The notion of CGMM via GGoF cores can also be extended in a variety of "grand" ways. Seven of the most promising are

1) The techniques of ROC analysis and Hotelling trace can be redefined with respect to the reduced assumption that the distributions are generalized projective Gaussian.
2) The intrinsic dimensionality of a distribution can be explored in terms of the dimensionality and extent of its GGoF core.

3) New methods for multidimensional scaling and data visualization can be defined based on the projection of a set of samples onto their core.

4) Higher dimensional GGoF cores can be extracted. In this manner, a broader range of problems can and should be attempted. In fact, the use of GGoF cores of the same dimensionality as feature space eliminates the continuous Gaussian distribution assumption; arbitrary distributions can then be represented.

5) The GGoF core can be incorporated into a "whole" model classification system. Whole model systems assign sample membership based on the simultaneous consideration of every class present.

6) Without modification the methods presented can be used for clustering unlabeled data instead of just for classification. Each non-overlapping GGoF core will correspond to a unique cluster of data in feature space.

7) GGoF cores can also be used in the creation of a hybrid supervised/unsupervised CGMM system. Such a system would be based on recent work using deformable linked loci for warping a medialness core based object model to an instance of that object. In the same manner, for example, a GGoF core based CGMM that was developed for representing one dialect of speech can be warped to represent an alternate dialect.