VALUE FOCUSED THINKING APPLICATIONS TO SUPERVISED PATTERN CLASSIFICATION WITH EXTENSIONS TO HYPERSPECTRAL ANOMALY DETECTION ALGORITHMS

THESIS

MARCH 2015

David E. Scanland, Captain, USAF

AFIT-ENS-MS-15-M-121

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-15-M-121

VALUE FOCUSED THINKING APPLICATIONS TO SUPERVISED PATTERN CLASSIFICATION WITH EXTENSIONS TO HYPERSPECTRAL ANOMALY DETECTION ALGORITHMS

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

David E. Scanland, BS, MS

Captain, USAF

March 2015

AFIT-ENS-MS-15-M-121

VALUE FOCUSED THINKING APPLICATION TO SUPERVISED PATTERN CLASSIFICATION WITH EXTENSIONS TO HYPERSPECTRAL ANOMALY DETECTION ALGORITHMS

David E. Scanland, MS

Captain, USAF

Committee Membership:

Dr. K. W. Bauer
Chair

Dr. J. O. Miller
Member

AFIT-ENS-MS-15-M-121

Abstract

Hyperspectral imaging (HSI) is an emerging analytical tool with flexible applications in many different target detection and classification environments, including Combat Search and Rescue, Military Intelligence, environmental conservation, and many more. Algorithms are being developed at a rapid rate, solving various related classification and detection problems under certain assumptions. At the core of these algorithms is the concept of supervised pattern classification, which trains an algorithm to data with enough generalizability that it can be applied to multiple instances of data. It is necessary to develop a logical methodology that can weigh attributes and responses and provide the analyst an output value that can help determine which algorithm should be used in a specific situation. This research focuses on the comparison of the overall quality of supervised learning classification algorithms (including Naive Bayes, Classification Trees, and Quadratic Discriminant) through the development, use, and analysis of a value focused thinking (VFT) hierarchy. This hierarchy represents a fusion of qualitative and quantitative parameter values developed with the use of elicited Subject Matter Expert a priori information. Parameters include a fusion of bias/variance values decomposed from both quadratic and zero/one loss functions, as well as a comparison of cross-validation methodologies and resulting generalization error. This methodology is then utilized to compare the aforementioned classifiers as applied to hyperspectral imaging data. The conclusions reached include a proof of concept of the credibility and applicability of the value focused thinking process to decisions for use of algorithm for different objectives.

Acknowledgments

I would like to express my heartfelt gratitude to my faculty advisor, Dr. Kenneth Bauer, for sticking with me throughout this whole process and inspiring me to keep pushing to find each new angle of analysis that could be uncovered. His humor and sociability helped me gain enough momentum to overcome my own inertia when I got in my own way. His profound knowledge of this subject is absolutely crucial to my success in this research effort as well as my completion of AFIT. I would also like to thank my reader Dr. J.O. Miller, as well as Trevor Bihl, Todd Paciencia, Joe Bellucci, Mike Gibb, Michael Mariotti, Greg Anderson, and the plethora of COA warriors that helped me get through this challenge. Every word of encouragement paved the road to completion and success. Send it in man!

David E. Scanland

Table of Contents

Page

Abstract iv
Table of Contents vi
List of Figures x
I. Introduction 1
Background 1
Methodology 4
Step 1: 5
Step 2: 6
Step 3: 6
Step 4: 6
Preview 9
II. Literature Review 10
Overview 10
Hyperspectral Data Analysis Algorithms 12
Collection of Hyperspectral Imaging Data 15
Radiance vs. Reflectance 19
Spectral Variability 20
Supervised Learning/Pattern Classification 23
Naïve Bayes Classifier 28
Quadratic Discriminant Analysis 29
Classification Trees 31
Confusion Matrix 36

Value Focused Thinking....38
Bias/Variance Dilemma....51
Friedman's Formulation....63
Bias and Variance for Classification....63
III. Methodology....71
Overview....71
Value Focused Thinking....72
Step 1: Problem Identification....72
Step 2: Creating the Value Hierarchy....73
Step 3: Developing Evaluation Measures....73
Step 4: Creating Value Functions....74
Step 5: Weighting the Value Hierarchy....75
Step 6: Generating Alternatives....75
Step 7: Scoring Alternatives....76
Step 8: Deterministic Analysis....76
Step 9: Sensitivity Analysis....76
Step 10: Conclusions and Recommendations....77
VFT Hierarchy....77
First Experimental Design....78
Classification Algorithms....82
Experimental Measures/Responses for the VFT Hierarchy....86
Cross-Validation....93
HSI Data....95

Factors for HSI Data Experiment 97
Application of Various Bias/Variance Frameworks for Classification 98
Conceptual Definitions 99
Bias and Variance for Regression (and TPF/FPF Values) 101
Domingos' Formulation 102
Definition 1- Main Prediction: 103
Definition 2- Bias: 103
Definition 3- Variance: 103
Definition 4- Noise: 104
Theorem 1 – Squared Loss: 104
Theorem 2 – Zero-One Loss: 105
IV. Results and Analysis 106
Value Focused Thinking 106
Step 1: Problem Identification 107
Step 2: Creating the Value Hierarchy 108
Step 3: Developing Evaluation Measures 109
Step 4: Creating Value Functions 109
Step 5: Weighting the Value Hierarchy 110
Step 6: Generating Alternatives 116
Step 7: Scoring Alternatives 116
Step 8: Deterministic Analysis 127
Step 9: Sensitivity Analysis 132
Step 10: Conclusions and Recommendations 133

V. Conclusions 134
Limitations 135
Suggestions for Future Research 136
Conclusions 136
Appendix A. Value Functions for Measures 138
Appendix B. Bias and Variance Comparisons 146
Appendix C. MATLAB Code 148
Appendix D. Quad Chart 155
Bibliography 156

List of Figures

Page

Figure 1. - Spectral Processing Algorithms (Shaw, 2002)..... 15

Figure 2. - Reflectance Spectrum for Vegetation and Soil (Manolakis, 2003)..... 16

Figure 3. - Representation of HSI Data (Manolakis, 2003)..... 17

Figure 4. - HSI Imaging Collection Process (Dube, 2009)..... 19

Figure 5. - Radiance and Reflectance Translation (Shaw et al., 2003)..... 20

Figure 6. - Target Detection Algorithm Issues (Manolakis, 2010)..... 21

Figure 7. - Two Dimensional HSI Representation (Manolakis, 2003)..... 22

Figure 8. - Supervised Learning Overview (Raschka, 2015)..... 25

Figure 9. - Testing, Training, and Validation (Dougherty, 2013)..... 26

Figure 10. - Quadratic Discriminants (Duda et al., 2001) 31

Figure 11. - CART Representation (Duda et al., 2001)..... 33

Figure 12. - CART Decision Boundaries (Duda et al., 2001)..... 34

Figure 13. - CART Decision Boundaries (Kuncheva, 2004)..... 34

Figure 14. - CART Decision Measurements (Dougherty, 2013)..... 36

Figure 15. - Confusion Matrix Terms (Sharma et al., 2009) 37

Figure 16. - Decision Thresholds and ROC Curve Representation (Dougherty, 2013) ... 38

Figure 17. - Value Focused Thinking Advantages (Keeney, 2009) 40

Figure 18. - VFT 10-Step Process (Shoviak, 2001)..... 45

Figure 19. - VFT Process for Evaluators and Warfighters (Bassham, 2006) 46

Figure 20. - Bassham's VFT Methodology (Bassham, 2006)..... 47

Figure 21. - ATR Value Hierarchy (Bassham, 2006)..... 48

Figure 22. - Warfighter Value Hierarchy (Bassham, 2006)..... 51
Figure 23. - Bias and Variance Comparisons (Fortmann-Roe, 2014) 53
Figure 24. - Bias and Variance per Model Complexity (Hastie et al., 2009) 54
Figure 25. - Bias and Variance per Number of Parameters (Dougherty, 2013) 55
Figure 26. - Representation of Boundary Bias (Dougherty, 2013)..... 65
Figure 27. - erf function values (Duda et al., 2001)..... 66
Figure 28. - Regression Bias and Variance (Duda et al., 2001)..... 68
Figure 29. - Classification Bias and Variance (Duda et al., 2001) 70
Figure 30. - VFT 10-Step Process (Shoviak, 2001)..... 72
Figure 31. - Categorical and Continuous Value Functions..... 75
Figure 32. - VFT Hierarchy 78
Figure 33. - Target and Background Distributions 81
Figure 34. - Naive Bayes Classification 82
Figure 35. - Classification Tree Example 83
Figure 36. - Quadratic Discriminant Analysis 84
Figure 37. - Euclidean Distance (Tomaselli et al., 2013) 85
Figure 38. - Mahalanobis Distance (Tomaselli et al., 2013)..... 86
Figure 39. - Computational Complexity Branch..... 87
Figure 40. - Classification Accuracy Branch..... 89
Figure 41. - Algorithmic Error Branch 91
Figure 42. - Parametric Bootstrapping (Shalizi, 2011)..... 92
Figure 43. - Non-Parametric Bootstrapping (Shalizi, 2011)..... 93
Figure 44. - k-fold Cross-Validation (Raschka, 2015) 94

Figure 45. - ARES Images 1, 2, 3, 1D (Orloff et al., 2000) 96
Figure 46. - ARES 1F, 2D, 2F, 3D10K (Orloff et al., 2000) 97
Figure 47. - ARES 3F, 4F (Orloff et al., 2000) 97
Figure 48. - VFT 10-Step Process (Shoviak, 2001) 106
Figure 49. - VFT Image Weighting Process 108
Figure 50. - Ease of Use Value Function 109
Figure 51. - Computation Time Value Function 110
Figure 52. - Global Weights - Computational Complexity 110
Figure 53. - Global Weights - Classification Accuracy 111
Figure 54. - Global Weights - Algorithmic Error 111
Figure 55. - Local Weights - Computational Complexity 112
Figure 56. - Local Weights - Classification Accuracy 112
Figure 57. - Local Weights - Algorithmic Error 113
Figure 58. - TPF/FPF Bias/Variance vs. Mahalanobis Distance 117
Figure 59. - Misclassification Rate vs. Target Pixel Percentage and Fold Number 118
Figure 60. - Jackknife Misclassification Rates 118
Figure 61. - TPF, FPF vs Pruning Level 119
Figure 62. - TPF, FPF vs. Leaf Size 119
Figure 63. - Domingos Bias/Variance Methodology 120
Figure 64. - Domingos' Boundary Error 121
Figure 65. - Computation Time Comparion 122
Figure 66. - FPF Measure Comparison 123
Figure 67. - K-fold Cross Validation Error Comparison 124

Figure 68. - Jackknife Cross Validation Error Comparison 124
Figure 69. - TPF Bias/Variance Comparison..... 125
Figure 70. - FPF Bias/Variance Comparison..... 126
Figure 71. - Domingos' Bias/Variance Comparison 127
Figure 72. - Hierarchy Values per Target Pixel Percentage and Mahalanobis Distance 128
Figure 73. - Hierarchy Values for 1% Target Pixel Pct and Short Mahalanobis Dist 129
Figure 74. - Hierarchy Values for 1% Target Pixel Pct and Long Mahalanobis Dist 130
Figure 75. - Hierarchy Values for 5% Target Pixel Pct and Short Mahalanobis Dist 130
Figure 76. - Hierarchy Values for 5% Target Pixel Pct and Long Mahalanobis Dist 131
Figure 77. - Hierarchy Values for 10% Target Pixel Pct and Short Mahalanobis Dist .. 131
Figure 78. - Hierarchy Values for 10% Target Pixel Pct and Long Mahalanobis Dist .. 132
Figure 79. - Classification Bias Local Sensitivity Analysis 132
Figure 80. - Ease of Use Value Function..... 138
Figure 81. - Computation Time Value Function..... 138
Figure 82. - TPF Measure Value Function 139
Figure 83. - FPF Measure Value Function..... 139
Figure 84. - Error Measure Value Function..... 140
Figure 85. - Accuracy Measure Value Function..... 140
Figure 86. - Sensitivity Measure Value Function 141
Figure 87. - Specificity Measure Value Function..... 141
Figure 88. - K-fold Cross Validation Error Value Function..... 142
Figure 89. - Jackknife Cross Validation Error Value Function 142
Figure 90. - TPF Bias Value Function..... 143

Figure 91. - TPF Variance Value Function 143
Figure 92. - FPF Bias Value Function 144
Figure 93. - FPF Variance Value Function 144
Figure 94. - Classification Bias Value Function 145
Figure 95. - Classification Variance Value Function 145
Figure 96. - CART Bias Comparison 146
Figure 97. - Naive Bayes Bias Comparison 146
Figure 98. - QDA Bias Comparison 146
Figure 99. - CART Variance Comparison 147
Figure 100. - Naive Bayes Variance Comparison 147
Figure 101. - QDA Variance Comparison 147

List of Tables

Page

Table 1. - Classification Algorithms 7

Table 2. - Data Experiment Factors 7

Table 3. - Hyperspectral Research Resources 11

Table 4. - Evaluators MOP's (Bassham, 2006) 49

Table 5. - Warfighter's MOE's (Bassham, 2006) 50

Table 6. - Types of Measures used in VFT 74

Table 7. - Classification Algorithm Alternatives 80

Table 8. - Factors used in Experiment 80

Table 9. - Measures for Computational Complexity and Classification Accuracy 86

Table 10. - Confusion Matrix 87

Table 11. - Confusion Matrix Formulae 88

Table 12. - Matlab Confusion Matrix Output 88

Table 13. - Measures for Algorithmic Error Response 90

Table 14. - ARES Image Factors 95

Table 15. - HSI Data Experiment 98

Table 16. - Local and Global Weights for Values 114

Table 17. - Local and Global Measure Weights 114

Table 18. - Global Tier Rankings 115

Table 19. - Global Measure Rankings 116

Table 20. - Color Representation in Tables 116

Table 21. - Computational Complexity Measures 121

Table 22. - Classification Accuracy Measures 122

Table 23. - Algorithmic Error Measures..... 123

Table 24. - Aggregated Hierarchy Values 128

VALUE FOCUSED THINKING APPLICATION TO SUPERVISED PATTERN CLASSIFICATION WITH EXTENSIONS TO HYPERSPECTRAL ANOMALY DETECTION ALGORITHMS

I. Introduction

Background

The No Free Lunch Theorem states that a comparison of classifiers for a classification task is largely dependent on the task at hand. Wolpert and Macready formalized this as

> All algorithms that search for an extremum of a cost function perform exactly the same, when averaged over all possible cost functions. In particular, if algorithm A outperforms algorithm B on some cost functions, then loosely speaking there must exist exactly as many other functions where B outperforms A (Wolpert and Macready, 1995).

A classifier that works in one case may outperform another classifier, but the other may outperform it in another case. Therefore, there are many factors and responses that go into determining which classifier is best in certain situations. Even if an optimal classifier did exist, it would be almost impossible to prove such a fact, as it would necessitate a vast amount of real-world data with many different characteristics to prove its optimality. Dimitris Manolakis states in his paper *Is There a Best Hyperspectral Detection Algorithm?*,

> Our main conclusion is that if we take into account important aspects of real-world hyperspectral imaging problems, proper use of simple detectors, like the matched filter and adaptive-cosine estimators, may provide acceptable performance for practically relevant applications. Are we certain that an undiscovered optimal detector does not exist? Probably not. However, even if such a detector were found, we may never have sufficient data to prove its superiority (Manolakis et al., 2009).

This task always necessitates an expert analyst stay in the loop to make any tactical decisions depending on the type of classification task that arises in the particular situation.

For this reason, in the Air Force, when analysts are observing remote sensing data in order to provide a decision maker the information that they need to make important real time decisions, they must fully understand the information that is given to them and must also have a

way to weigh different methodologies that infer information out of the data that they have. Specifically, when assessing hyperspectral imaging data collected from remote sensors aimed at an operational scene in order to detect targets of interest, the analyst must understand which tool will give them the best results amongst a plethora of different objectives and criteria. This situation naturally leads to the idea and use of Value Focused Thinking (VFT) or Multi-Objective Decision Analysis (MODA) to assess the quality of these different anomaly detectors in different situations.

Hyperspectral imaging (HSI) analysis is a discipline that allows analysts to collect data about the environment that is of their interest in a unique way that takes advantage of all of the information that is contained in the Infrared and Visible portions of the Electromagnetic Spectrum. Every material that exists in this universe is comprised of a unique spectral fingerprint that can be extracted if one knows how to look for it. Much like in the discipline of statistics in general, where data is collected in a certain methodological way and then analyzed and tested in order to discover an inference about this data that describes the truth of some underlying population, HSI allows the collection Electromagnetic (EM) information along the spectral dimension (partitioned by wavelength) for the inference of the type of material in each of a certain amount of pixels in the spatial dimension of a sensor based on that sensor's resolution. This naturally leads to the use of pattern classification algorithms to detect the classes of these materials based on their spectral decomposition, using the EM information that is treated as the features, or predictors of these classes. If an initial data collection experiment is conducted, and a truth set of information is formed for that specific image, the classification algorithm that is being used can be trained in order to optimize the discriminant of the features that can separate the targets, or anomalies, from the background data.

There are many issues prevalent when attempting to measure the performance of classification algorithms and many competing responses that can be used in order to train the algorithm to detect the certain targets of interest. The quality of any type of methodology is a difficult concept to define. Robert Pirsig states in his book, *Zen and the Art of Motorcycle Maintenance*,

> 'What's new?' is an interesting and broadening eternal question, but one which, if pursued exclusively, results only in an endless parade of trivia and fashion, the silt of tomorrow. I would like, instead, to be concerned with the question 'What is best?,' a question which cuts deeply rather than broadly, a question whose answers tend to move the silt downstream (Pirsig, 1974).

To know the methodology that is truly best helps improve knowledge and thus allows the creation of new methodologies in an improved direction that will provide the most utility.

One issue belonging to the question of "what is best?" is the contextual information that is present within the scene that is being sensed. What types of targets are of interest? To what degree do they blend in with the surrounding background pixels? Are there different targets that are of interest that can be detected and separated from the background that need to be assessed and weighted per their importance and criticality? How alike is the current image that is being analyzed and the other images that will be analyzed in the future, and can our classification algorithm be trained robustly enough to account for these differences and still provide the analyst the information that is needed to make the right inferences? How difficult are the algorithms to create, manipulate, use and maintain and can the average analyst use them in every situation or does there need to be a Subject Matter Expert (SME) in the loop to help conduct these tasks? How long does it take the algorithm to make a correct classification and does this length of time match up appropriately to the operational situation that it is being used in? It is clear that there is

a lot of information to think about and a lot of questions to be asked in order to make the right decision to which algorithm is most beneficial.

Value Focused Thinking (VFT) is used in order to analyze a space of many competing objectives, which is the situation that occurs in HSI Anomaly Detection, and it weighs each competing objective appropriately based on the sets of measurements and values that are inherent to those objectives. This allows the influx of prior information from decision makers (DMs) and subject matter experts (SMEs) that can update our state of information for the particular situation's requirements. Without this addition of SME input, the comparisons that are made will be based on frequentist assumptions of probability and likelihood, which assumes that only the data collected in the experiment, along with the assumptions of asymptotic normality, can be used to make a decision for which of the alternatives are better in that case. This can be flawed as in some cases, some of these measures will be valued higher than in other cases, and the overall decision that would be made based on expected values will not apply to every single case as a whole. Using a hierarchical Value Focused Thinking approach allows us to focus on the values that are important to us in that specific situation so we can collect useful actionable information that can be used to form and assist important decisions in a wartime environment. This research effort is a development of a VFT framework in the context of an HSI anomaly detection data collection and decision making effort using different classifiers as our alternatives.

Methodology

There is an inherent lack of formalism in the literature for deciding upon which algorithm is most useful for classification in which situation. Due to this vacuum of knowledge, a study can be conducted on a small scale of situations that can lend information about which of a set of algorithms should be used in that situation, along with the opinions of the Decision Maker. This

is akin to the situation of when a certain weapon must be used in theater to deter a particular threat. Not every weapon is equal in every situation, and a *de facto* analysis must be conducted in order to assess the validity of that weapon in that situation. Viewing HSI data classification as one of those weapons, this analytical effort will attempt to set up an organized hierarchical comparison using multiple responses, values, and measurements of interest that are formulated under the supervision of a typical Subject Matter Expert/Decision Maker in order to make *ad hoc* decisions of which classifier to use in which situation. This will allow the analyst at least a blueprint of a methodology that can be manipulated in order to organize all of these responses in a logical manner instead of just performing guesswork.

In summary, the approach that will be undertaken in this research effort will be comprised of a sequence of experiments utilizing various pattern classification techniques for two-dimensional and multi-dimensional HSI anomaly detection and analysis. The process is illustrated in the following list of steps.

Step 1:

Decide upon and list responses or objective measurements that define the "quality" of the HSI Anomaly Detection Algorithm. These responses are broken into three areas; the first area considers the computational effort required to use the algorithm, which can be thought of as the user's satisfaction or dissatisfaction. The second area is an analysis of the confusion matrix, which represents how well the classifier is labeling data points as targets or backgrounds after it has been sufficiently trained. The third area is an analysis of the training and validation performance of the algorithm, in terms of how it performs when subjected to various training and testing data sets.

Step 2:

Values for responses will be entered into a VFT hierarchy which is internally weighed and assessed using the input from the SMEs and DMs. This VFT analysis will provide a single value for the quality of the algorithm in light of the fusion of both statistically and subjectively garnered information across all performance measurements.

Step 3:

The single value computed and recorded from the VFT Hierarchy will be compared with the results of each individual performance measurement. This will be an empirical verification of the differences in the recommendations of the hierarchy over the individual measurements. This could provide both additional inferences about the algorithms as well as an assessment of the different methodologies of performing algorithm analysis and comparison.

Step 4:

Post-processing will be done by first analyzing the hierarchy values per significant factor levels. Additional processing can be done by weighting the hierarchical values by the percentage of pixels in the image to simulate the weighing of the classifiers by operational scenario.

For the first step, two multivariate normal distributions will be randomly generated representing a background class and a target class. Different supervised learning classification algorithms will be utilized to assess the effects of various factors on the classification behavior and performance, and in turn, this will allow us a transparent view at how the VFT hierarchy is performing. All reasonable permutations of factors will be used to develop a multitude of unique combinations. These combinations are considered different unique images for this first experimental stage. The factors that are in play include the type of classification algorithm, the

Mahalanobis distance between the centroids of the two distributions, the covariance matrices of distributions, and the percentage of target pixels to overall pixels. The following is the table of factors and their levels.

Table 1. - Classification Algorithms

Alternative Classification Algorithms
Quadratic Discriminant Analysis
Naïve Bayes Classification
Classification Trees

Table 2. - Data Experiment Factors

Mahalanobis Distance	Target Covariance Matrix	Background Covariance Matrix	Percentage of Target Pixels
Long >10	TCM1	BCM1	1%
Short <10	TCM2	BCM2	5%
	TCM3	BCM3	10%

Each individual factor is a representation of what is most likely to occur in a basic imaging anomaly detection problem. After each VFT Hierarchy value is collected, a post-processing adjustment for this value will be computed by weighing the number of target pixels in the image. This will allow the weighing mechanism to account for changes in the quality of the algorithm due to the unique scenario that it is being used for. In actuality, there is a large difference in using an algorithm in passive, non-time sensitive situations, such as analyzing crop distributions in a field, and those of more active and urgent situations, such as those found in Search and Rescue and Military settings.

The responses that will be generated and input into a VFT hierarchy include a measurement for computational complexity in terms of the difficulty to perform this basic task,

and the computational time that it takes to perform the task. The second response will be a selection of the False Positive Fraction (FPF) and True Positive Fraction (TPF) that would typically be seen in a Response Operator Characteristic curve (ROC), these same values but in terms of a Specificity and Sensitivity framework, or these values under an Error and Accuracy framework. This allows the analyst to choose a framework to work under in order to reduce any ambiguity in the decision chain from analyst to Decision Maker. In this research, the TPF and FPF values will be used. The third component will be comprised of error under the framework of Cross Validation weighed against the framework of Resubstitution Error, as well as the Decomposition of the Mean Squared Error (MSE) function for classification utilizing bootstrapping to weigh the Bias of the algorithm when calculating TPF and FPF to the Variance of the algorithm when computing these same values.

The second experiment will use these same classification algorithms but with HSI data generated from a data collection effort using the simulated Airborne Reflective Emissive Spectrometer (ARES) Forest Radiance I and Desert Radiance II data collection experiment images developed from the Hyperspectral MASINT Support to Military Operations (HYMSMO) program using the Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor. Six of the HYDICE ARES images in the AFIT Sensor Fusion Library will be used to assess these classification algorithms. The same responses will be calculated as in the first experimental effort and these responses will be input into the same VFT hierarchy. Similar analysis will be conducted using this Hierarchy, with focus being placed on how the VFT changes when the algorithms are used for a more complex dataset.

Preview

Chapter 2 serves as a survey of the background information that is needed to fully understand the context and content of this research effort. It will delineate the current knowledge that it contained in the literature of HSI Anomaly Detection algorithms as well as the concept and application of Value Focused Thinking. This chapter also describes the responses that are of interest and the various classification schemes that are analyzed within this research effort. Chapter 3 gives insight into the approach of the analysis within the framework of HSI Anomaly Detection, Value Focused Thinking, and Experimental Design. It will illustrate the sequential nature of the effort that is performed and the comparisons that will be made. The chapter will also outline the Value Focused Thinking process and the steps involved to develop the hierarchy, develop the value functions, and assess the measurements. Chapter 4 contains the results of the analysis that is undertaken to compare the classification algorithms. Any additional insights that come to light from these comparisons are contained within this chapter. Chapter 5 details the conclusions and inferences that have come about from this research as well as a listing of contributions to the field of HSI Anomaly Detection and any suggestions for additional research as follow on studies that can be conducted using the results of this analysis.

II. Literature Review

Overview

The realm of hyperspectral imaging is a burgeoning field that has exploded over the last twenty years. The reason for its growth is its universal application in many fields, including medicine, law enforcement, military, homeland security, and developing Graphic User Interfaces such as Google Earth (Dube, 2009). Many attempts have been made to improve the performance of algorithms under the considerations of various assumptions, including non-linearity, non-independence, and non-normality of the background and target spectrum distributions. Borghys, et al. states "HSI anomaly detectors differ in the way the background is characterized and in the method used for determining the difference between the current pixel and the background" (Borghys et al., 2007). The constraints and difficulties found within this application cause the procession and evolution of algorithms to continue at a rapid pace. Many thesis research efforts have been focused on improving different algorithms and implementing algorithms in order to show incremental improvement. Some of these algorithms include classical, finite-target, and mixture-tuned matched filters; Reed-Xiaoli (RX) anomaly detector; orthogonal-subspace, adaptive-cosine estimator; and subspace, kernel-matched subspace, and joint subspace detectors. Along with these algorithms, there have been various methods developed for data treatment tasks such as feature extraction and selection for dimensionality reduction, background-clutter modeling, end-member selection, and radiance-versus-reflectance domain processing (Manolakis et al., 2009). Table 3 highlights a selection of references that are found within the HSI domain for common types of algorithms and types of data treatment tasks.

Table 3. - Hyperspectral Research Resources

Types of Algorithm	References	Types of Data Treatment Tasks	References
Classical Matched Filters	(DiPetro et al., 2012) (Nasrabadi, 2014) (Shi et al., 2010)	Feature Extraction	(Liao et al., 2013) (Lunga et al., 2014) (Kang et al., 2014)
Finite-Target Matched Filter	(Manolakis et al., 2002) (Schaum et al., 2004) (Stocker et al., 1997)	Feature Selection	(Li et al., 2011) (Serpico et al., 2001) (Yu et al., 2002)
Mixture Tuned Matched Filter	(Mundt et al., 2005) (Lentilucci et al., 2005) (Thompson et al., 2013)	Background-Clutter Modeling	(Stein et al., 2002) (Kasen et al., 2004) (Burr et al., 2006)
Reed-Xiaoli Anomaly Detector	(Banerjee et al., 2006) (Borghys et al., 2011) (Williams et al., 2013)	End-Member Selection	(Winter, 2009) (Du et al., 2008) (Plaza et al., 2004)
Orthogonal-Subspace Estimator	(Chang et al., 2011) (Acito et al., 2010) (Bioucas-Dias et al., 2012)	Radiance-versus-Reflectance Domain Processing	(Shaw et al., 2002) (Staenz et al., 1998) (Lentilucci et al., 2009)
Adaptive-Cosine Estimator	(Manolakis et al., 2013) (Frontera-Pons et al., 2012) (Pieper et al., 2011)		
Subspace Detector	(Guo et al., 2011) (Zhang et al., 2010) (Gholizadeh et al., 2012)		
Kernel-Matched Subspace Detector	(Chen et al., 2011) (Wang et al., 2013) (Gu et al., 2011)		
Joint Subspace Detector	(Eismann et al., 2009) (Zhang et al., 2010) (Borghys et al., 2012)		

Due to the rapid expansion of knowledge in this field, it is a necessary to create a methodological framework to balance the values that are utilized when comparing these

algorithms. Each individual algorithm must be compared to other algorithms under the same assumptions, as it is a difficult task to enumerate each assumption and compare each algorithm across different levels of reality and mathematical rigor. As Dimitris Manolakis (Manolakis, 2009) states, “It is both time consuming and difficult for designers of hyperspectral imaging systems to navigate through the existing literature to choose a detector or decide if a certain level of performance can be expected” (Manolakis et al., 2009). The following few pages represent an elicitation of knowledge for various subjects across the algorithmic and response variable domains in order to create a representation of topics that are relevant to creating a logical framework for comparison. These include overviews of the HSI domain, Value Focused Thinking, Supervised Learning algorithms, the Bias-Variance tradeoff, and the Confusion Matrix.

Hyperspectral Data Analysis Algorithms

Figure 1 outlines the basic tasks that are found in HSI image analysis. There are three main classes of algorithms that have been developed, each for different aims of utilizing the information that is collected from the HSI imaging sensor. The first, Target Detection, is what this research effort is concerned with. Target detection is the classification of pixels within an image as either target or background pixels. Target detection algorithms can be characterized in two separate groups, including Spectral Anomaly Detection Algorithms and Spectral Matching Detection Algorithms. Anomaly Detection Algorithms do not need the a priori spectral signatures of the target pixels to work. When comparing the pixel with either the local or global background, any pixel that does not have the same spectral composition is declared a target. While this is a desirable property, it is limited by the fact that it cannot separate anomalies that are man-made, natural, or targets of interest. Atmospheric compensation is not a necessary piece

of information that is required for anomaly detection. It is the aim of the algorithm to draw a discriminating boundary that can separate the target pixels and background pixels. This means that the task is a binary classification task, with the two classes being target and background. The other type of detection algorithm is the spectral matching algorithm which does need a priori information about the target of interest in order to distinguish whether it is present in the scene. Pixels are measured in terms of how correlated their spectrums are with known target spectrums. These known spectrums for the targets can either be taken from truth libraries or from other pixels where the targets are identified (Manolakis et al., 2009).

The two other types of algorithms are Change Detection and Classification. Change Detection is the analysis of HSI data in the spatial and temporal domains in order to detect whether and how a scene changes over those two dimensions. This allows for the observation of movement, which is especially important in military applications, when subjects of interest could be attempting to camouflage their movement to avoid detection. Classification is the expansion of the Target Detection task into multiple class labels in order to detect and record the difference in materials using a dictionary with recorded spectral information for specific materials or using pixels in the scene to characterize the spectral information. Shaw states, “Formally, classification is the process of assigning a label to an observation (usually a vector of numerical values), whereas detection is the process of identifying the existence or occurrence of a condition” (Shaw et al., 2002). Each of these three algorithm types can be further split into the domains of dealing with pure pixels, where the materials of interest occupy full pixels in the image, and mixed pixels, where there are a percentage of different materials in the same pixel (Shaw, 2002).

Additionally, Dimensionality Reduction and Unmixing are data processing techniques that deal with issues in the data. Dimensionality Reduction deals with the idea of the “Curse of

Dimensionality". The "Curse of Dimensionality" is the property that classification tasks become increasingly difficult when more and more dimensions, or features, are added into the problem (Friedman, 1996). When HSI sensors collect 210 bands of information about the reflectance of the materials in a scene, there is a lot of information that can be reduced due to correlations between the features. Dimensionality Reduction is the process of removing the excess information that increases the computational cost of the analysis while maintaining the amount of information that is used to differentiate between the target and the background. Shaw states, "Dimensionality reduction leads to significant reductions in computational complexity and also reduces the number of pixels required to obtain statistical estimates of a given accuracy" (Shaw, 2002). Usually, Dimensionality Reduction is performed using Principal Components Analysis. Unmixing is the process of looking at pixels with more than one class of material within them and using an estimation of the amount of those materials to aid in the distinction between the classes. Unmixing is important for images that do not have spatial resolution high enough to perfectly distinguish target from background, which is almost all of the cases when collecting images in real-world scenarios. Unmixing will not be approached within this research effort.

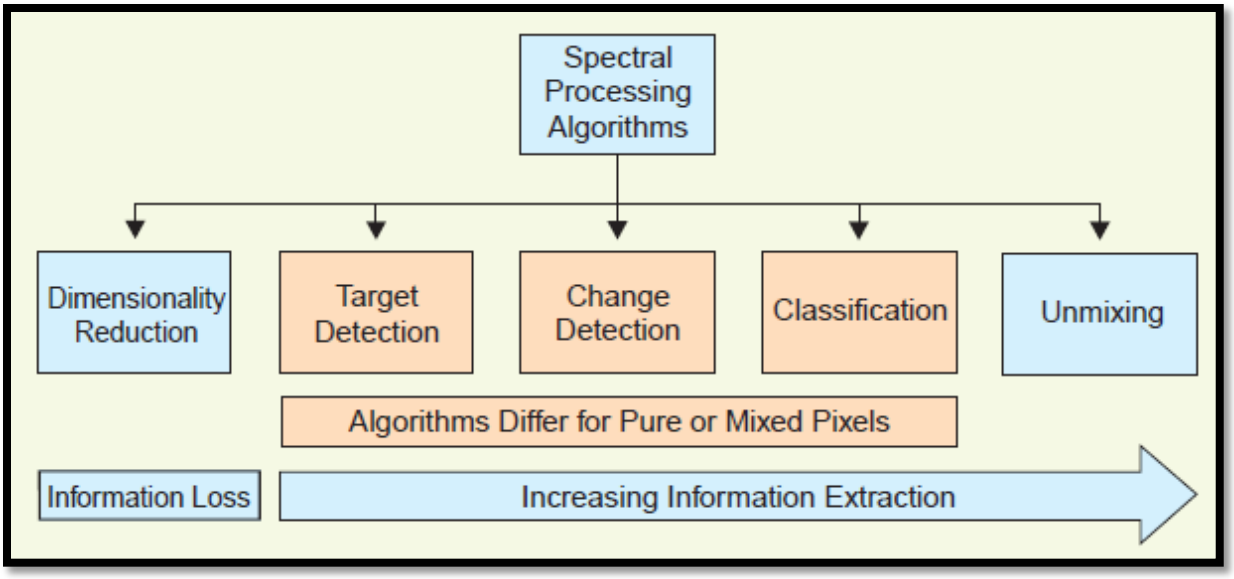


Figure 1. - Spectral Processing Algorithms (Shaw, 2002)

Collection of Hyperspectral Imaging Data

Fingerprinting has been a method of identifying and classifying individuals for many years in detective and forensics work. This example of classification has become a cliché in media, including in films and television shows. The process is simple and is often unaccompanied by any type of statistical algorithm. The only items needed for this type of classification are the fingerprints at the scene and a database, or truth set, of fingerprints that they can be matched to with some degree of certainty. This matching allows the detective the ability to discriminate potential matches that correspond with those unique fingerprints from those who have different patterns, and ultimately, result in substantive evidence that can be used to convict a person of a crime.

Hyperspectral imaging is the fingerprinting of the remote sensing and imaging world. Instead of unique, literal fingerprints, the methodology is used to perform a type of pattern matching using the way that the unique material reflects, absorbs, and emits electromagnetic energy that it is exposed to from various sources, including most prevalently, the illumination

from the sun. This phenomenon of reflectance, absorbance, or emittance and the translation of it into a unique signature that can be analyzed to tell it apart from other materials is captured by sensors. These sensors are focused on a scene in order to collect a set of images as if they were a stack of playing cards, with each card corresponding to a unique spectral bandwidth in the visible, near-infrared (NIR), and mid-infrared (MIR) portions of the electromagnetic spectrum. Figure 2 represents the spectral reflectance signatures of green vegetation, soil, and dry vegetation within the visible and NIR portions of the spectrum.

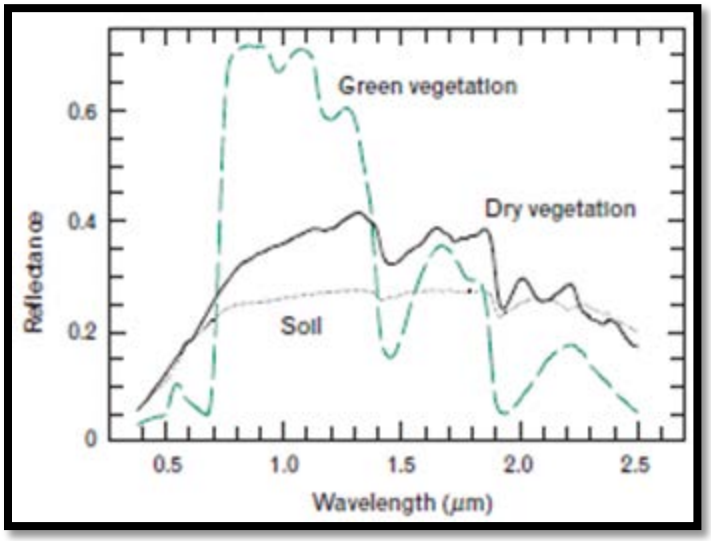


Figure 2. - Reflectance Spectrum for Vegetation and Soil (Manolakis, 2003)

Each of these playing cards can also be analyzed and interpreted in the more intuitive spatial dimension in order to discern which pixels contain targets or anomalies that may be of interest to the analyst and ultimately the warfighter and the decision maker. The analysis of the images in the spatial dimension helps to create the truth database that each spectrum has to be compared to in order to perform discrimination and detection of each individual pixel.

The size and representation of the pixels on the ground depend on the spatial resolution as well as the collection mechanism of the sensor, with each of these sensors commonly

connected to aircraft or satellites making passes over the specific area of interest on the ground. Some pixels may represent miles of area and thus contain many materials that mix the spectrums of the pixels, making it more difficult to tell materials apart and thus necessitating more advanced algorithms with higher computational cost and effort to distinguish each material from these mixed pixels. This scenario is analyzed using the Unmixing Algorithms discussed previously. Some pixels may represent only small swaths of land of a few square feet in area, which contain unique signatures that could represent metal from tanks, skin from individuals, or the organic spectrums of trees and shrubs. It is these pixels that are in fact hiding vectors of spectral information in the 3rd spectral dimension of what is known as a data cube, which is the collection method and data interpretation of choice in HSI analysis. The representation of the data cube is seen in the Figure 3. The image on the left is a representation of the cross-section of reflectance values in the spectral domain for an individual pixel that will be classified as either a target or background. The image on the right is the spatial representation of the image for an individual spectral band.

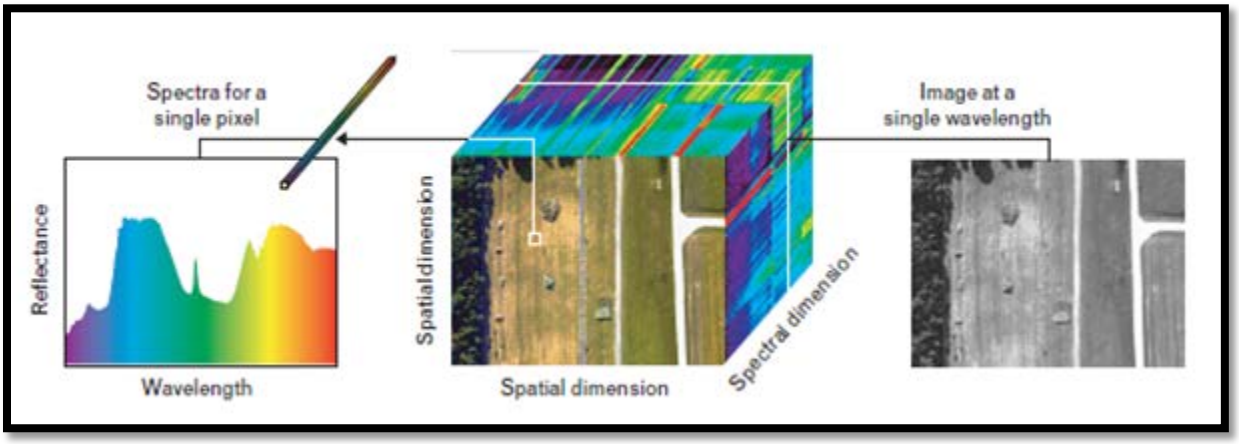


Figure 3. - Representation of HSI Data (Manolakis, 2003)

While the data cube seems like some far-out, new approach to collecting data, it is really just a cognitive representation of data that could still be unfolded into the typical data matrix,

representing the independent variables, also known as predictors or features in classification nomenclature, and the dependent response variable vector format that is common in most fields of statistical science. The pixels only stand in as place-keepers that represent the index number of this unfolded matrix format. The response variable in the HSI Anomaly Detection methodology is the class of the pixel, often representing a binary 1 or 0, to represent what are considered target pixels that are of interest to background pixels that the target pixels must be distinguished from. Target/Anomaly Detection algorithms help perform the distinguishing as they set up a measurement parameter(s) that can be used to determine whether the pixel is considered an anomaly or part of the background. For anomaly detection, spectral information is not considered part of the *a priori* knowledge set for the data, but only information whether the pixel represents a target or part of the background. The algorithm then uses a distance measurement or a distinguishing measurement between the Pixel Under Test (PUT) and the distribution of the background to determine whether the pixel is a target or background pixel (Borghys et al., 2012).

Figure 4 shows a spectral decomposition of soil, water, and vegetation for reflectance data in a scene, collected from a sensor on an aircraft. As seen in the figure, each spectrum in the Visible and NIR spectrums is significantly different from one another and can be distinguished using the appropriate algorithm. For anomaly detection, the background in this situation could be represented by any of the soil, water, or vegetation spectrums, while the target would most likely be the buildings or roadways in the image. However, these are arbitrary designations based on input of the analyst and Decision Maker.

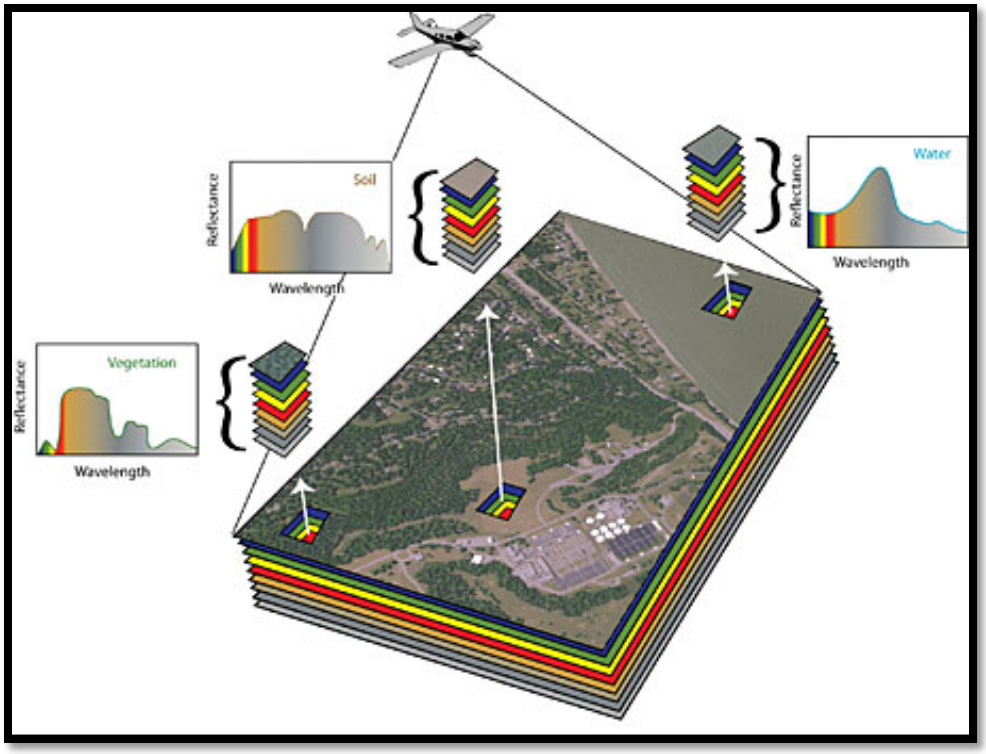


Figure 4. - HSI Imaging Collection Process (Dube, 2009)

Radiance vs. Reflectance

An important distinction in the collection of HSI data is the difference between Radiance and Reflectance data. The sensor observes and digitizes the radiant flux, or radiance, that enters the sensor's aperture. For each ground pixel, the radiance is composed of both the illumination that comes directly from the sun's rays and the amount that the material reflects back into the sensor. These can be separated as the radiation reflected from the pixel of interest itself, the radiation reflected from the surface surrounding the pixel of interest and scattered in the air, and the radiance the occurs due to the photons scattered without ground contact (Manolakis et al., 2009). These measurements are different for each individual wavelength. Other factors that creep into the equation include the angle of the sun, the viewing angle of the sensor, the solar radiance from atmospheric scattering, the illumination from reflected light of other materials, shadows in the scene, and atmospheric scattering, along with biases from the

sensor (Shaw, 2002). Therefore, pre-processing must be accomplished in order to compare apples to apples. Usually, this includes taking the pure radiance data and performing atmospheric compensation to determine reflectance data, which is then used in the data processing applications and then in the unmixing and detection algorithms. This process is seen in Figure 5.

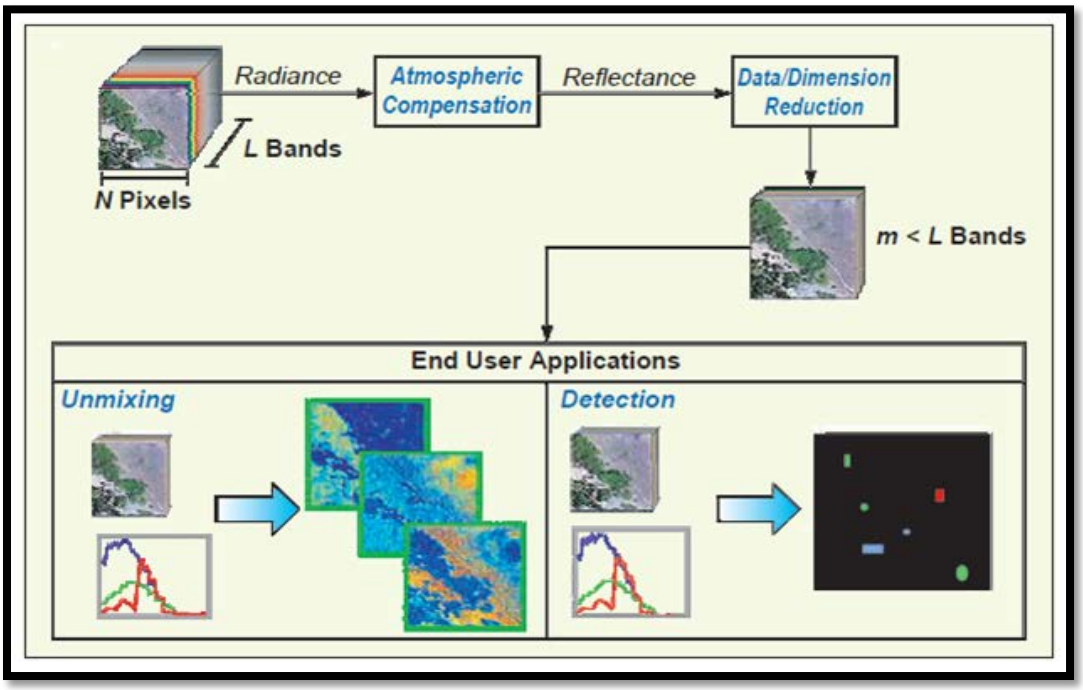


Figure 5. - Radiance and Reflectance Translation (Shaw et al., 2003)

Spectral Variability

One of the difficulties in anomaly detection is the variability of the target material's spectrum in the wavelength domain and the interaction with these spectrums with the spectrums of those of background materials. These inherent variabilities are a result of atmospheric attenuation and scattering, sensor resolution, and slight changes in material composition (Manolakis et al., 2009). This causes the problem to go from the deterministic domain and adds noise to go to a stochastic domain. Additionally, the resolution of the sensor in the spatial domain needs to be appropriate for the situation, so targets of interest can be fully separated from the background for each individual pixel. If this is not done, mixed pixels occur, and

methodologies such as target fill factors must be used to distinguish the components of the pixel. These issues are highlighted in Figure 6.

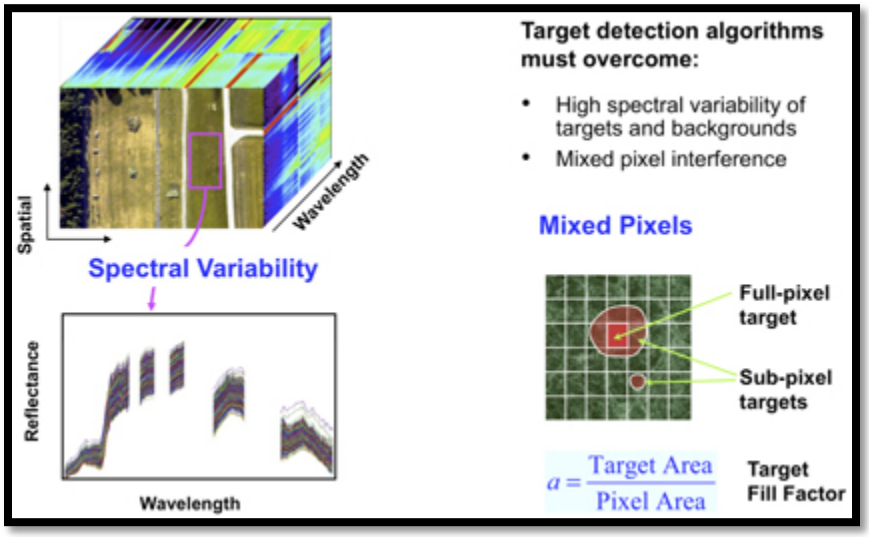


Figure 6. - Target Detection Algorithm Issues (Manolakis, 2010)

Often times, the features at the unique spectral bands are modeled as normal distributions that are correlated with the other spectral features of each of the pixels. Therefore, the collection of pixels for each class can be described as multivariate normal distributions with certain means, or centroids, and covariance matrices that describe the covariance, which can be normalized as the correlation of the factors in the vector for each pixel. Statistical inference methodologies using normal distributions have been studied and used extensively in the literature of classification and anomaly detection algorithms due to their mathematical representations and robustness of performance (Manolakis and Shaw, 2002). Manolakis and Shaw state that

> Algorithms based on normality assumptions are used to derive many detectors due to their usefulness in many practical applications, the theoretical intuitiveness of their operation and performance, and their use for the discovery and development of algorithms for nonnormally distributed HSI data (Manolakis and Shaw, 2002).

Figure 7 displays an example of class separation in a cross-section of two spectral bands, 620 μm and 960 μm. This example is a false-coloring image where each color represents a different class of object in the image. It can be seen that some objects are easier to classify than others, with various levels of heterogeneity.

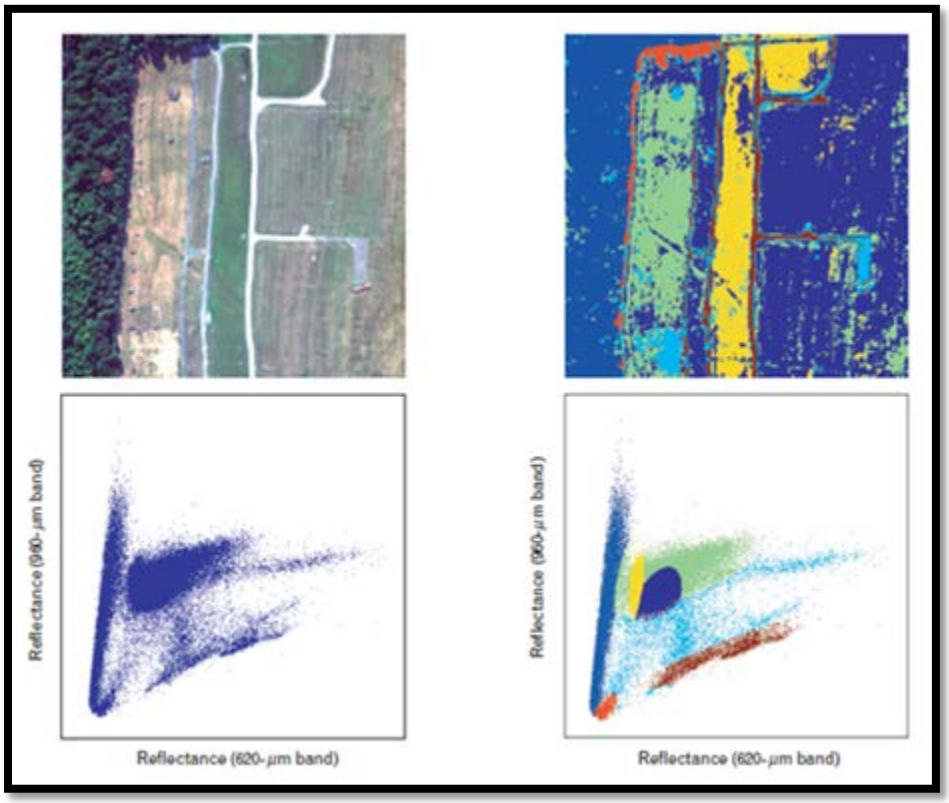


Figure 7. - Two Dimensional HSI Representation (Manolakis, 2003)

The overall methodology of the development and comparison of detection algorithms hinges on the ability to accurately model the spectral variability that is inherent to the target and background distributions (Manolakis et al., 2009). This variability is a product of the size of the target in the scene and its mixing with background within pixels, the environmental conditions present in the scene, sensor noise and resolution, and the stochastic component of the error within the spectra of the target and the background.

An alternative approach that will not be studied in this research is the geometric approach which treats the spectrum as a vector that varies in an M-dimensional subspace of the data space ($M<p$), where p is the number of spectral bands. This approach treats the spectrum as a linear combination of vectors that constitute the subspace of the variability. These vectors are known as endmembers and can be taken from a library of previously collected data for the specific material or obtained using eigenvectors from the correlation matrix for the spectral bands (Manolakis et al., 2002).

Basic methods for the whole domain of supervised learning and classification will be detailed in the following section. These algorithms represent the classification of data points with various amounts of assumptions.

Supervised Learning/Pattern Classification

The following is a brief discussion of the methodology of Supervised Learning. Within the realm of supervised learning, the class labels are known up front and are used to build and improve the classifiers. Since within the datasets used in this research the pixels that represent targets and those that represent background are known *a priori*, this research deals primarily in Supervised Learning. This is in contrast to unsupervised learning algorithms, in which the class labels are unknown, and instead, analysis is done to estimate and separate classes using their intrinsic qualities. Unsupervised learning is primarily done with various clustering algorithms.

The process of supervised learning is comprised of the original collection of the data, which in HSI is the sensor collection of HSI data from an aircraft or satellite. This raw data gets pre-processed by filling in values for missing data or extracting features using methods such as Principal Component Analysis to reduce the size of the feature matrix. There could also be some initial work done due to the images or datasets not being in a standard form. This would be the

case when sensors collect images at different angles and in different weather conditions. This reduced feature matrix is then sampled by splitting some of the data in a training set and some in a test set. Theses training and test sets are then put through an additional round of pre-processing in order to further reduce any redundant information (Raschka, 2014).

The resulting training set is placed within a learning algorithm to train it to make correct decisions about target and background splits. Once the learning algorithm is sufficiently trained, certain hyperparameters are developed and optimized to assess model quality. These hyperparameters are quality assessments of the algorithm that are independent to any learning that is accomplished from the training sets of data. These parameters include bias and variance estimates that ensure adequate generalization of the algorithm. Once these parameters are optimized, the model is usually kept as a representative model. Throughout this training process, cross validation is accomplished to split the available data into groups for training and some for testing and finally validation. Training allows the model to learn the data and draw adequate decision boundaries. Testing data is done to ensure that the model is generalized to fit other sets of data without propagating too much bias (Raschka, 2014).

Post-processing is done by assessing the model using a confusion matrix that is comprised to true positive, true negatives, false positives, and false negatives. These values are also manipulated into other values that can be used to assess certain probabilities of classification performance. The main goal of using a confusion matrix is to develop robust measurements that can be utilized in many different situations with many different assumptions, including prior costs and probabilities. For that specific instance of the interaction between the model and the training set, the total number and percentages of classification rates are recorded and used to get an overall interpretation of model accuracy. Each one of these measurements is prone to its own

bias and variance and caution must be heeded when attempting to make any overarching logical conclusions using this type of post-processing assessment. From this step, the model can then be tested recursively on new data, which helps the analyst optimize parameter values, or it can be finally validated on a separate set of data. When this tuning and refinement is completed, the model can then be used to make predictions on brand new, real-world data, with some confidence that the model is performing well (Raschka, 2014). Figure 8 captures this process.

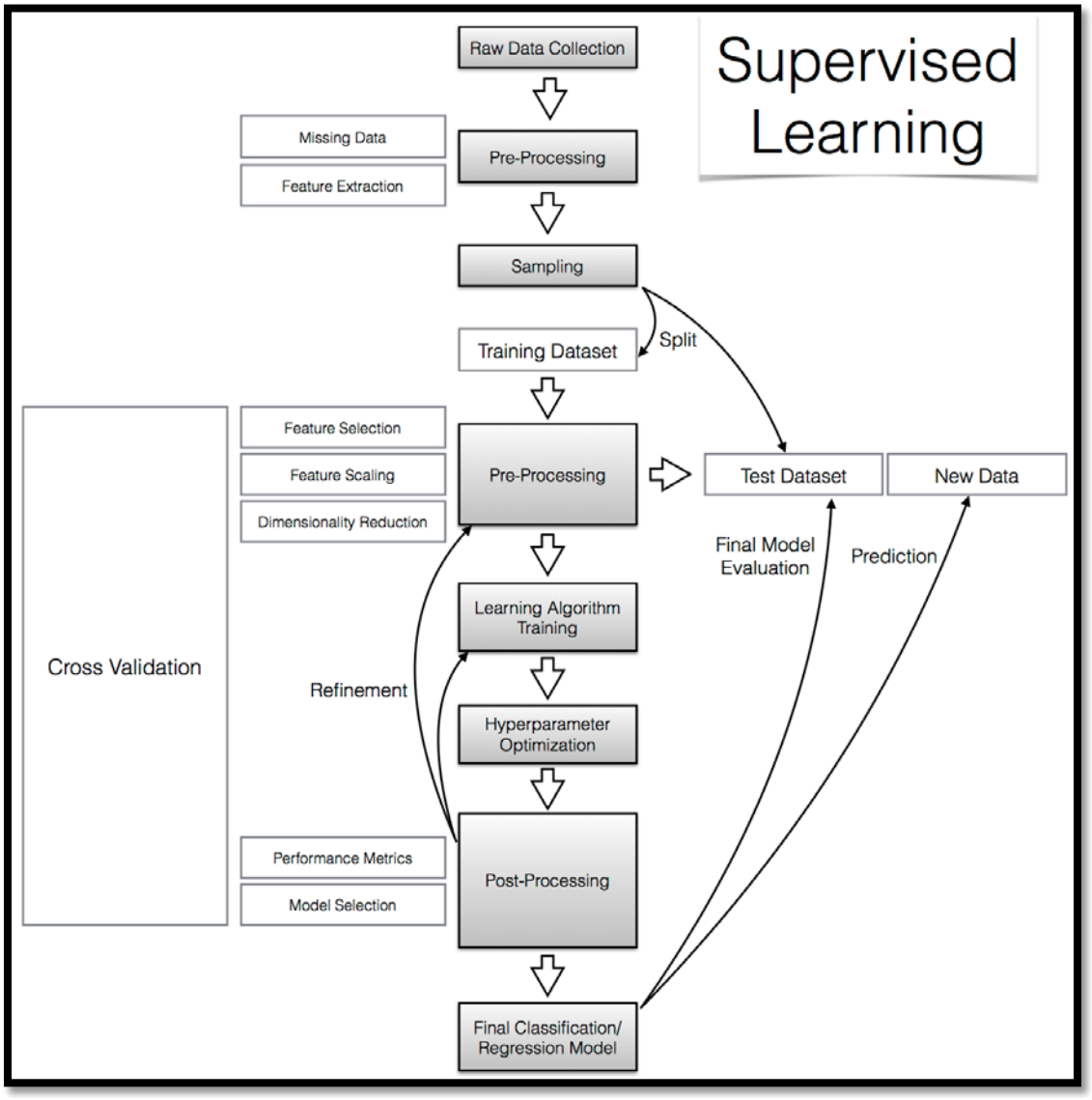


Figure 8. - Supervised Learning Overview (Raschka, 2015)

The process of training, validating, and testing the model is seen in Figure 9. Error, consisting of bias, variance, and noise, is propagated when using the model developed from the training set to model the data in the validation set. Several instances of the model at various degrees of complexity are developed, and the validation set error (after integrating all forms of error) is minimized to find the optimum complexity. The training and validation sets are thus utilized for assessing the final model, which is then assessed against a test set, which measures the performance that is expected for that particular model specification. The test set error components are then integrated and used as a guideline for the amount of error present in the model.

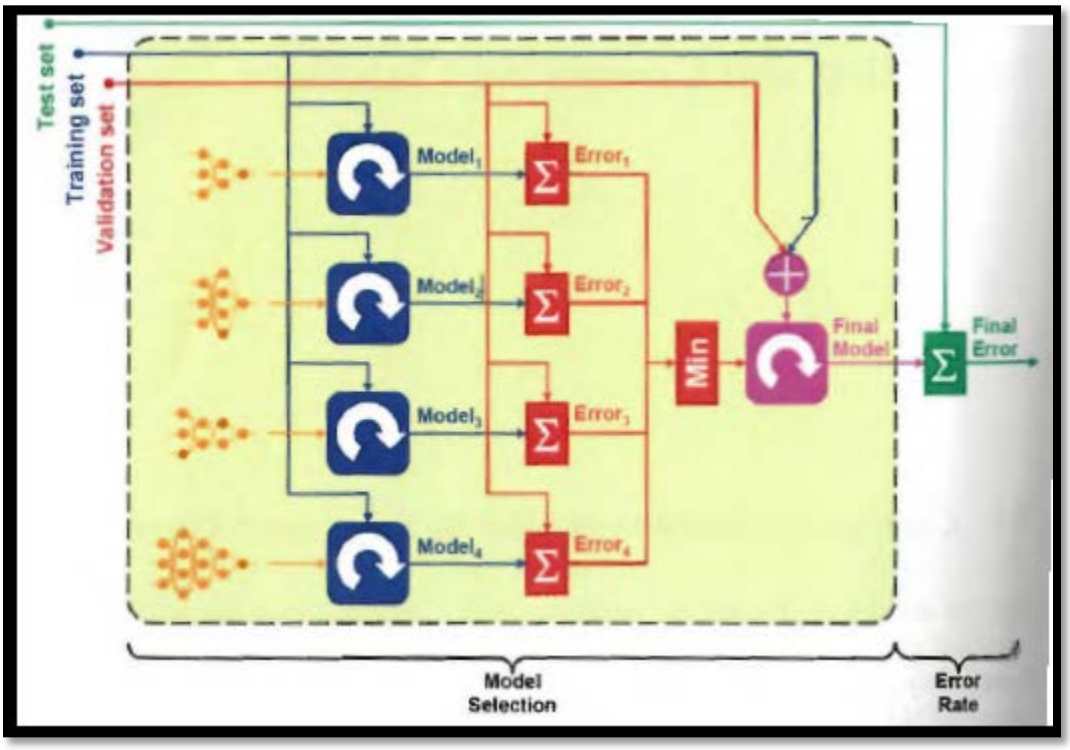


Figure 9. - Testing, Training, and Validation (Dougherty, 2013)

Within the realm of anomaly and detection algorithms, likelihood ratio tests are often used to test whether the pixel under test is part of the background or part of a different, target distribution. The basic formulation of the likelihood ratio test is based on the null hypothesis that the pixel is from the background distribution and the alternative hypothesis that it is part of a different target distribution. This is formulated as a ratio of probabilities (Dougherty, 2013):

$$\Lambda(x) = \frac{f_1(x|target\ present)}{f_0(x|target\ absent)} = \frac{f_1(x|H_1)}{f_0(x|H_0)} \quad (1)$$

where

x represents the vector of the pixel under test

$\Lambda(x)$: Likelihood ratio value

$f_1(x|H_1)$: likelihood of observing x under the target present hypothesis (H_1)

$f_0(x|H_0)$: likelihood of observing x under the target absent (background) hypothesis (H_0)

If the value of $\Lambda(x)$ is over a threshold that is chosen by the analyst, the pixel is considered a target. If it is below the threshold, it is considered part of the background. In many systems, the goal is to maximize the probability of detection while keeping the probability of false alarm as low as possible. This is known as the Neyman-Pearson (NP) criterion (Manolakis et al., 2002).

The multivariate Gaussian distribution is often used for hyperspectral image classification. This distribution is used to model target distributions that are full pixel targets. This distribution is also used to model the background pixels in the scene, and there may often be a mixture of several multivariate Gaussian distributions inherent within the background. The hypothesis test that is used in this situation is based on the following hypotheses:

$$H_0: x \sim N_p(\mu_b, \Sigma_b) \quad (2)$$

$$H_A: x \sim N_p(\mu_t, \Sigma_t) \quad (3)$$

where

x represents the vector of the pixel under test

N_p is the pdf of the multivariate Gaussian distribution

μ_b, Σ_b are the mean and covariance matrix of the background distribution

μ_t, Σ_t are the mean and covariance matrix of the target distribution

Naïve Bayes Classifier

A naïve Bayes classifer is a discriminant function that is solely based on using Bayes' rule with the assumption that each of the features that are used within the target and background distributions are independent. Bayes rule is formulated as the following (Dougherty, 2013) (Duda et al., 2001):

$$Posterior(probability) = \frac{likelihood * prior(probability)}{evidence} \quad (4)$$

This formulation is equivalently,

$$P(Target\ Present|x) = \frac{p(x|Target\ Present) * P(Target\ Present)}{p(x)} \quad (5)$$

The posterior probability in the target/anomaly detection case is the probability that the pixel is a target given the features that are used for classification. Likelihood has the same interpretation as before, which is the likelihood that the pixel is part of the target distribution. This is equivalently interpreted as when all other things being equal, the category, target or background, for which the likelihood is larger, is more likely the true category. The prior probability is a measurement

of the knowledge that we have that we can predict *a priori* that the pixel is either part of the target or the background. Usually this is estimated from the number of pixels that are actually targets and the number that are actually background. The evidence is largely ignored in this formulation and is only a scale factor that states how frequently we will measure a pattern with the individual feature value and it ensures that the posterior probabilities sum to one (Duda et al., 2001).

This formulation assumes that the features are all independent in the scene and the classification is done simply by assigning the pixel to either target or background depending on the maximum *a posteriori* (MAP) probability. This means that each feature has a conditional probability of the predicting the class, and due to independence, the posterior probability is calculated by multiplying all of the probabilities for each individual feature together for that specific class. This resultant probability constitutes the likelihood. This is then multiplied by the prior distribution. Therefore, the decision is to select that the pixel is a target if the following holds (Dougherty, 2013):

$$p(x|Target\ Present) * P(Target\ Present) > p(x|Target\ absent) * P(Target\ absent) \quad (6)$$

Dougherty discusses the implications of the independence assumptions:

> Despite the fact that far-reaching independence assumptions are often inaccurate, the naïve Bayes classifier works well in many real-world situations. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality. Like all probabilistic classifiers under the MAP decision rule, it arrives at the correct classification as long as the correct class is more probable than any other class; hence, class probabilities do not have to be estimated very well. In other words, the overall classifier is robust enough to ignore serious deficiencies in its underlying naïve probability model" (Dougherty, 2013).

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis is used to build a Quadratic Detector in cases when the covariance matrix of the background does not equal the covariance matrix of the target

distribution. This discriminant function is quadratic due to the quadratic term still being present in the function. For two features, the discriminant will be ellipses, circles, parabolas, hyperbolas, lines or multiple lines (Dougherty, 2013). The likelihood ratio value is the following:

$$\Lambda(x) = \frac{|\Sigma_b|^{1/2}\exp\left[-1/2(x-\mu_t)^T\Sigma_t^{-1}(x-\mu_t)\right]}{|\Sigma_t|^{1/2}\exp\left[-1/2(x-\mu_b)^T\Sigma_b^{-1}(x-\mu_b)\right]} \quad (7)$$

The logarithm of this function yields:

$$y = D(x) = (x-\mu_b)^T\Sigma_b^{-1}(x-\mu_b) - (x-\mu_t)^T\Sigma_t^{-1}(x-\mu_t) \quad (8)$$

This is a comparison of the x vector under test of the Mahalanobis distance between the target and background distributions. Figure 10 shows the resulting decision boundaries for various constructs of the two Gaussian distributions that are used in the certain example. Due to the shapes of the elliptically contoured distributions, the boundaries could be straight lines, ellipses, circles, or hyperbolas, which are, in fact, all quadratic boundaries.

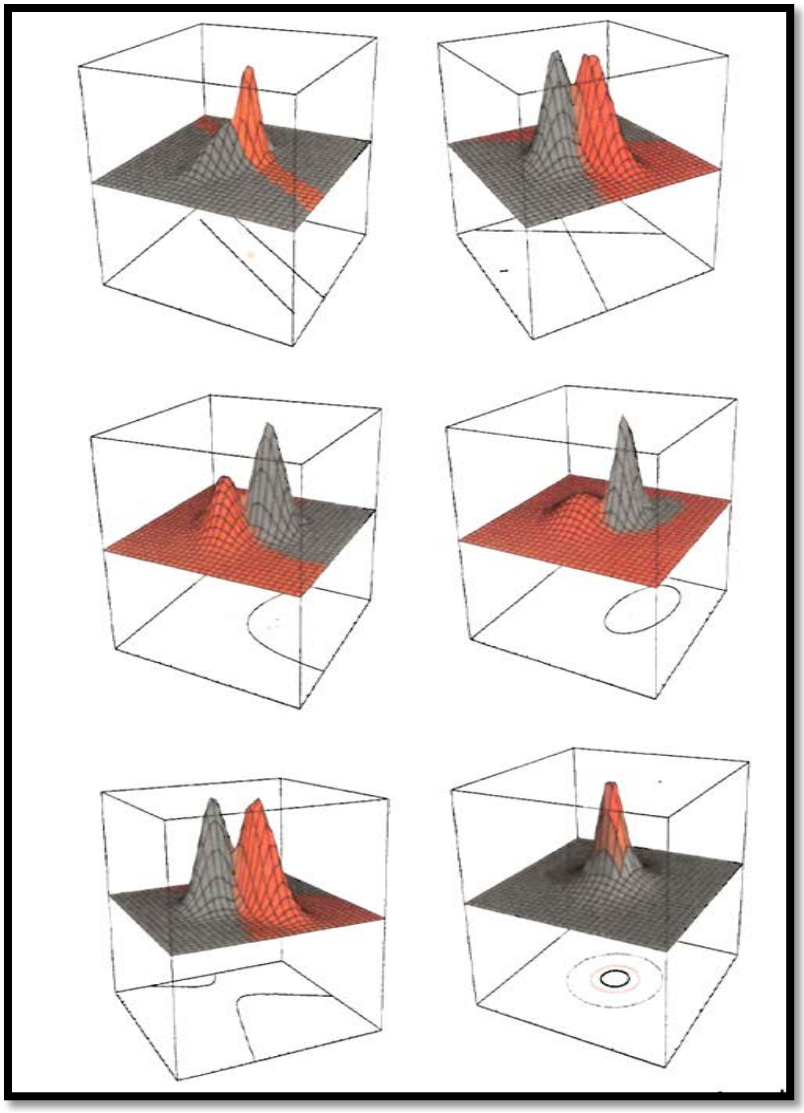


Figure 10. - Quadratic Discriminants (Duda et al., 2001)

Classification Trees

Classification trees are structures comprised of decision nodes that allow the analyst the ability to divide the training data set into groupings based on binary splits at each node. The starting node is denoted a root node that is considered to be the parent of every other node. Branches are formed by splitting the data at each individual feature in a recursive fashion. When the tree is built to the actual class of the response variable, or until some hyper-parametric threshold is reached, the tree reaches the leaf nodes. The tree is updated during the process of

fitting as it decides the best splits of the data based on the most significant features. The main advantages of the classification trees are that they are easy to use and interpret, which means that analysts can easily put them into operation without needing the complex knowledge necessary to explain and use other classification algorithms. Most often, the feature space that is analyzed with Classification Trees is comprised of categorical factors that have no direct interpretation of distance between one another. For example, there is no direct interpretation of the distance between category levels “Blue” and “Red”. Questions at each node can be asked to determine the correct state of nature for each of the response classes. These questions help the analyst understand what comprises each class in terms of attributes, and which attributes are the most important to explicitly describe the categories. At each node, a decision must be made to determine whether the node is a finalized leaf node based on the distribution of classes at that node, or whether another splitting criterion should be used to split into additional branches. Additionally, the tree structure is very receptive of Subject Matter Expertise, which can be used to narrow down the decision space.

The typical Classification Tree structure is seen in Figure 11, consisting of the root node that contains all of the points in the dataset. At this point, the most discriminating rule is used to split the data into two sets. This is whether or not the data in this case, the fruit, is green. This process is reiterated at these two separate nodes, using a value of a feature distribution that separates the data into two groups using a distance or information metric. When the process hits a certain threshold and can no longer be split, or each of the nodes contains values from only one class, the process is terminated. In this case, the largest leaf nodes contain only two fruit.

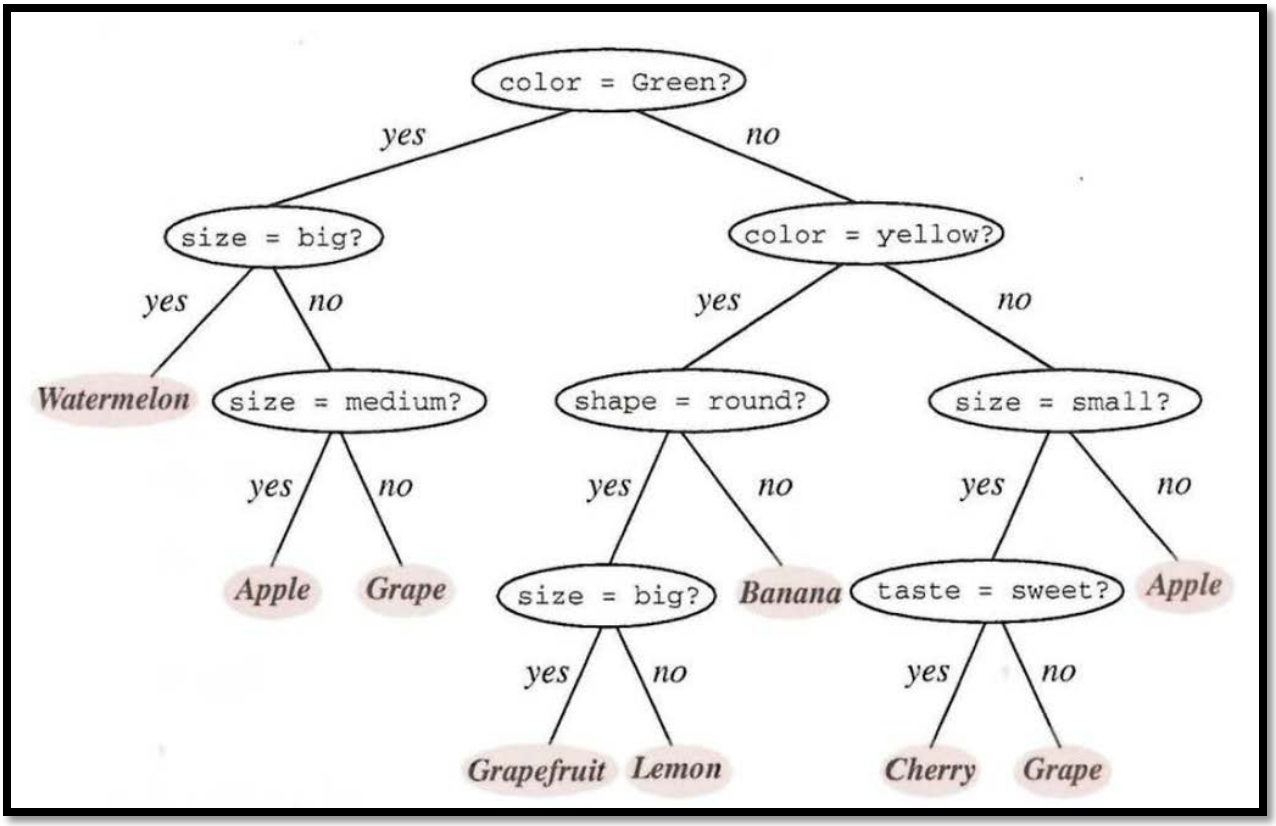


Figure 11. - CART Representation (Duda et al., 2001)

Figure 12 represents the types of decision spaces that are developed when creating and implementing trees. These spaces are necessarily perpendicular to the feature axes, as at each split, the question being posed is a binary decision that separates the classes in some proportion. However, any decision space can be approximately estimated by growing the tree as much as necessary. There is an inherent bias/variance tradeoff when creating these trees, as growing the tree too large is considered over-fitting to the data in the training set, and under-fitting the tree results in bias in that all classes are not specifically or accurately separated into classes. The proper size and complexity of the tree is computed by using pruning techniques that optimize certain parameters.

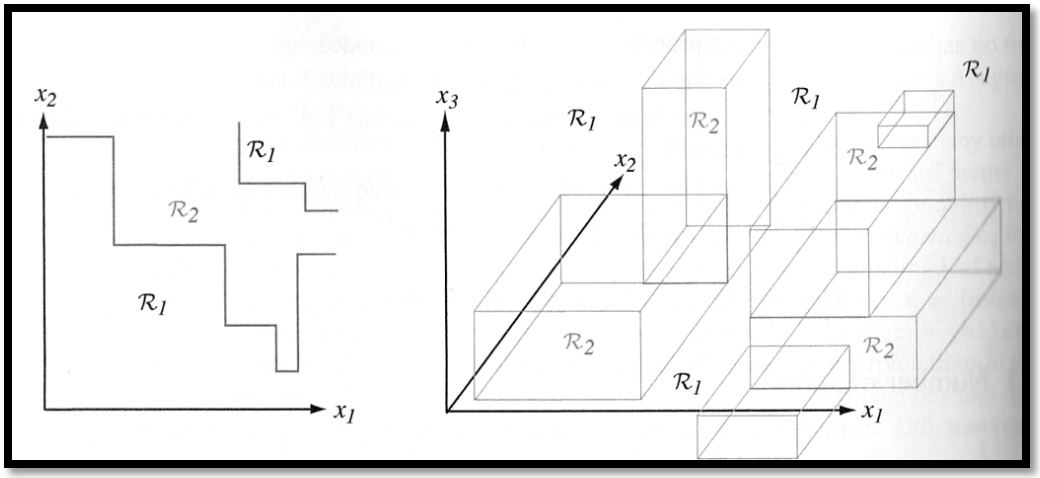


Figure 12. - CART Decision Boundaries (Duda et al., 2001)

Figure 13 is an additional representation of the orthogonal binary decision space that separates class labels in a rectangular grid like pattern. If more and more of these grids were overlaid on an image or dataset, the grid could roughly approximate any class distribution. (Kuncheva, 2004).

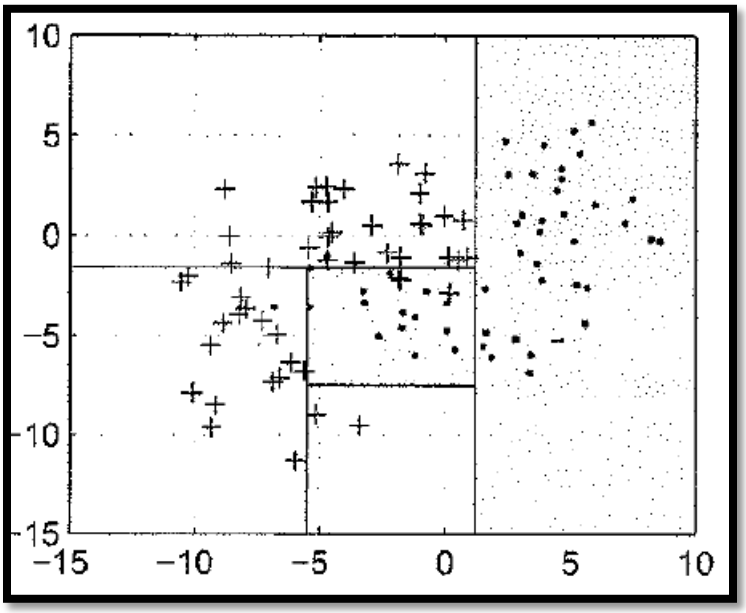


Figure 13. - CART Decision Boundaries (Kuncheva, 2004)

In order to create the best possible tree, there must be measurements of how each set of nodes is separating the classes as much as possible, so each final leaf node is as discriminatory as possible with a higher proportion of one class or another. The amount of mixing of class labels that each leaf mode contains is known as impurity, with a leaf node that has the same proportion of one class as it does the other having the most impurity, while a leaf node with only one class is at 100% purity, or 0% impurity. The various measurements of impurity are seen below (Duda et al., 2001).

Entropy impurity:

$$i(N) = -\sum_{\mathrm{j}} \mathrm{P}(\omega_{\mathrm{j}}) \log_2 \mathrm{P}(\omega_{\mathrm{j}}) \tag{9}$$

Gini impurity:

$$i(N) = \sum_{i \neq j} P(\omega_i) P(\omega_j) = \frac{1}{2} * [1 - \sum_{j} P^2(\omega_j)] \tag{10}$$

Misclassification impurity:

$$i(N) = 1 - \max_{j} P(\omega_j) \tag{11}$$

where

$P(\omega_j)$ is the prior proportion of class ω_j at the node

A representation of the various impurity measurements for a two-class problem is seen in Figure 14. As the proportion of one class to another approaches 0.5, the impurity measure reaches its maximum, which for Entropy is 1.0, while it is 0.5 for Gini and Classification Impurity. As the proportions get closer to 0.0 or 1.0, the impurity measurements tend towards 0. In Figure 14, the variable p is the proportion of data points that fall into some class for the node being analyzed. As the proportion of points at the node reach a uniform distribution of 0.5, each

of the measurements are at their highest levels. As the purity of class at the node increases (or decreases), each of these measurements will approach 0. The nodes with the lowest impurity measurement will be selected for the split, as they contain the most information, as the largest proportion of one single class will be represented by that node split.

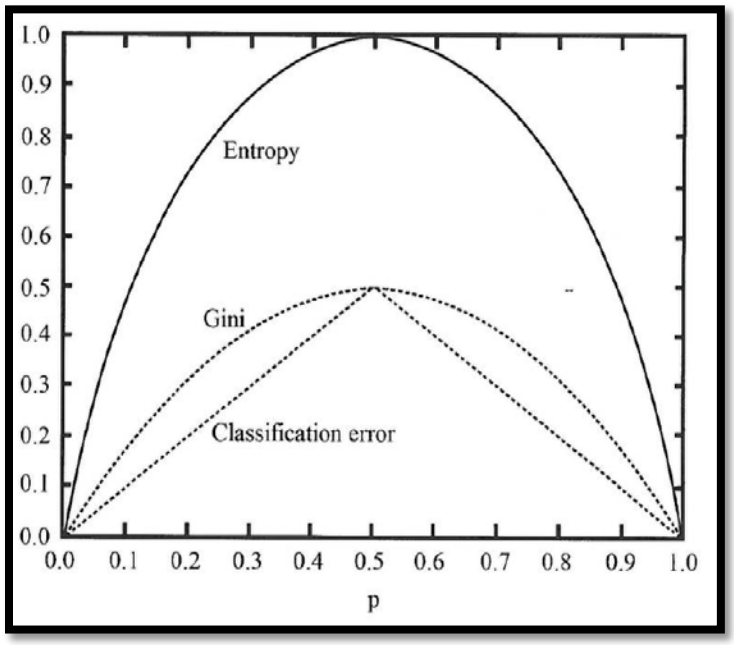


Figure 14. - CART Decision Measurements (Dougherty, 2013)

Confusion Matrix

A confusion matrix is the means in which a model can be assessed for accuracy of assigning data points to the correct classes for a binary decision classification. This decision is comprised of positives and negatives, usually with positives meaning some sort of target of interest, while negatives meaning some sort of background population that is not particularly of interest. Figure 15 is a representation of the typical confusion matrix and the resulting measures that can be derived.

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test Outcome Positive}}$
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test Outcome Negative}}$
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	**Specificity** = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	

Figure 15. - Confusion Matrix Terms (Sharma et al., 2009)

The confusion matrix is naturally extended to a graphical representation of distributions for the two class problem for each individual feature. Figure 16 is a representation of a single feature and the class distribution associated with that feature. The blue distribution is arbitrarily labeled the negative distribution and the red distribution is labeled the positive distribution. The area in blue is the probability that a data point in the negative population is correctly classified as a negative, which is calculated as the True Negative Fraction, and the area in red is the probability that a point belonging to the positive class is classified as a positive point, which is calculated as the True Positive Fraction. The light red area represents the case of a point belonging to the positive class being classified as a negative point, which is calculated as the False Negative Fraction, and the light blue area is the case where a truly negative data point is classified as belonging to the positive class distribution, which is the False Positive Fraction. The results for each of these percentages are seen in the matrix alongside the distributions. The TPF and FPF values are plotted within a Receiver Operating Curve (ROC), with each particular instance along the curve being calculated for some particular threshold or parameter value. The

Area Under the Curve (AUC), is a useful measurement of the accuracy of the classifier for thresholds and parameters of interest, with the line associated with a random guess of positive or negative going from the bottom left of the plot to the upper right. As the classifier becomes more accurate, the AUC value will approach unity (Dougherty, 2013).

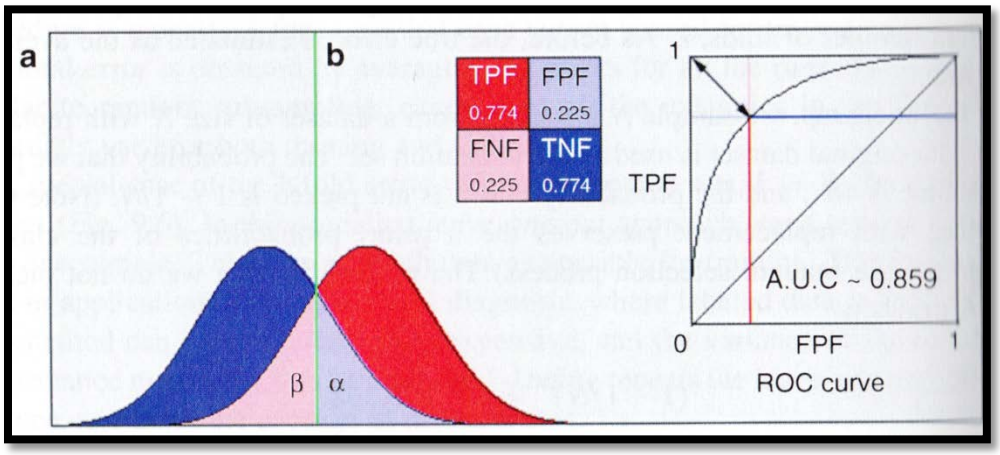


Figure 16. - Decision Thresholds and ROC Curve Representation (Dougherty, 2013)

Value Focused Thinking

The concept of Value Focused Thinking (VFT), was developed by Ralph L. Keeney at the University of California in order to break the intuitive trend of focusing on alternatives when making decisions, and thus trying to fit the options to the objectives and not the objectives to the options. Keeney considers the key goals that are used to develop the foundations of correct decisions to be values. He states, “Values are fundamental to all that we do; and thus, values should be the driving force for our decision making. They should be the basis for the time and effort that we spend thinking about decisions, but this is not the way it is” (Keeney, 1996). This is similar to the idea of jumping to conclusions without the necessary logic in place to form the basis of why the conclusions are valid in the first place. By forming this type of structural argument for the decisions that you make, new alternatives can be synthesized, and sometimes,

new values can be deduced from this logic, which makes the process iterative. Keeney believes that values should always be the first place to start.

Keeney expands on the delineation of alternative-focused thinking from value-focused thinking by arguing that alternative-focused thinking is more of a way to solve decision problems, while value-focused thinking goes beyond this realm and helps identify desirable decision opportunities and create new alternatives. He believes that there are three main differences in these perspectives. He states, “

> First, significant effort is allocated to make values explicit. Logical and systematic concepts are used to qualitatively identify and structure the values appropriate for a decision situation. Second, this articulation of values in decision situations comes before other activities. Third, the articulated values are explicitly used to identify decision opportunities and to create alternatives (Keeney, 1996).

From this statement, it is clear that he values structure in the decision making process that allows the optimization of inference from all of the information that is presented *a priori*.

Figure 17 lists the advantages of thinking about values over thinking about alternatives. The advantages range from the improvement of communication and relationships between the people involved in the decision, and the improvement of information collection, to helping create and evaluate alternatives, as well as discovering strategic policies and objectives that may have been hidden before. The main point of this process is that alternatives are only one product of keeping the values in focus and there is a plethora of other positive byproducts that help the organization.

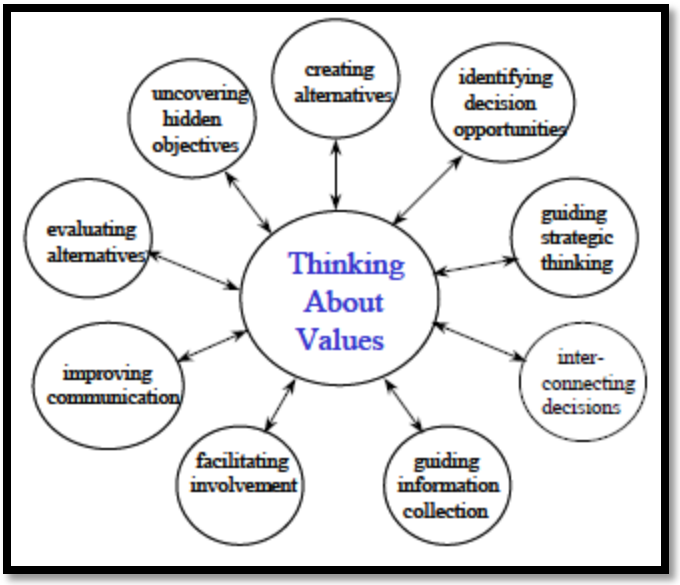


Figure 17. - Value Focused Thinking Advantages (Keeney, 2009)

Keeney discusses his view that decisions are complex ideas that are structured using multiple objectives, with each objective being a statement of some end that is desired to be reached in the context of the decision. This means that each objective is dependent on the decision context that it is analyzed and made in, the object that is being decided upon, and some delineation of the preference that a decision maker has, which is used to optimize the final result of the parameter that is extracted from the decision. There is also a difference between the terms 'fundamental objectives' and 'means objectives' which are both used in a decision making context. Fundamental objectives are the actual ends to the means that are valued in the context, while means objectives are met in order to achieve those ends. The broadest objectives in any organization are considered strategic objectives, which help meld all of the other decisions made by the organization.

In order to understand the true meaning of these objectives, many times the analyst must interview the Decision Maker (DM) or Subject Matter Expert (SME) to drill down to the logical statements and axioms present in the objectives. If there is confusion about what the objectives

actually represent, this confusion could propagate to the choice of alternatives and other decisions. This type of 'devil's advocate' analysis is also useful to discover other important objectives that had not been thought of previously. Additionally, it is important to rank and weight these objectives as some of the time and resources spent reaching one may be out of proportion with the actual impact and importance these objectives have on the overall decision making. By changing the distribution of weights from a uniform distribution to one that weighs more important objectives higher, a more realistic representation of the quality of each individual alternative that is applied to the decision can be achieved. All objectives should be listed, ranked, and a percent weight of importance should be distributed. Analyzing this list of objectives could help determine if some group of objectives are out of proportion with others or which ones need to be ranked higher than others based on strategic objectives.

Keeney discusses the steps within the value assessment as thus,

> The value assessment comprised several separate tasks: listing the objectives, distinguishing between means objectives and fundamental objectives, identifying measures for the objectives, and prioritizing them. The results of each task helped us to articulate company values and use these to suggest decision opportunities that might be worthwhile to pursue (Keeney, 1996).

This demonstrates the ability for this process to synthesize new ideas and create even more decision opportunities that were not known *a priori.* Many types of different surveys can be utilized beyond just interviewing Decision Makers and SMEs to extract information. These include surveys to any type of stakeholder, including, often, the general public, the employees or users of the alternatives, or anyone else that could be affected by the decision. Keeney explains,

> The strategic objectives of an organization can guide the identification of decision opportunities that enhance both the likelihood of achieving those objectives and the degree to which the objectives are achieved. This process, part of value-focused

> thinking, helps to put the decision-maker in control of the decisions being faced rather than leave that control to others and to happenstance" (Keeney, 1996).

In order to develop correct alternatives and make correct decisions, the analyst must be proactive in the process and should not wait for knowledge to reveal itself.

There is often confusion about how to list objectives and which ones should be used to help the analyst provide input to the Decision Maker. VFT contains many different procedures that assist in the compiling of objectives, categorizing the objectives as means or ends and logically ordering them, using the objectives to help discover or create new alternatives, and finally to understand new opportunities within the decision making process. It is necessary to poll the decision maker by asking for a comprehensive list of objectives under the assumption that there are no constraints limiting or preventing the fulfillment of the objectives. It is also important to then ask what the objectives would be after some amount of assumptions. Keeney states,

> Often one begins to think hard about a decision situation only after some alternatives become apparent. Articulating the features that distinguish existing alternatives provides a basis for identifying some objectives. For example, in considering alternative sites for an airport, one feature that differentiates the alternatives might be the disruptions to citizens due to high noise levels. This suggests the obvious objective of minimizing disruption from noise. You might ask respondents to list desirable and undesirable features of alternatives and use these to stimulate thought about objectives (Keeney, 1996).

The raw list of objectives that is generated from this procedure should be analyzed to correctly align each to either means objectives or fundamental objectives. If the objective is an essential reason for interest in the overall situation, the objective is a fundamental objective, while if it is just a means of accomplishing some other objective, it is a means objective, and the additional objective should be also assessed for importance and type. Specification must be done

to logically decompose the objective into its different parts, which could also lead to additional objectives. Keeney explains,

> Suppose the CEO of a service firm identifies one objective as 'to minimize nonproductive time spent by employees'. To better understand this objective, you might ask the executive to be more specific, or to list characteristics of nonproductive time. You might ask how nonproductive time occurs and whose nonproductive time is of concern. All of the responses should help specify the objective (Keeney, 1996).

Creating alternatives can prove to be a difficult task for various reasons. One reason is that many different types of alternatives could be left of an initial list. This is because there is often a need for analysts and decision makers to quickly find a limited set of alternatives and start working towards assessing those alternatives without expanding the list and taking the time to understand what is not on the list. There is also an anchoring effect that occurs due to the dependence that new alternatives have on previously listed alternatives, and each alternative will be within some radius of the other in terms of originality and scope. Most of the new alternatives will only be small tweaks of the previously deduced alternatives, and true originality is left in the minds of the analysts and DMs. Keeney argues, "Focusing on the values that should be guiding the decision situation removes the anchor on narrowly defined alternatives and makes the search for new alternatives a creative and productive exercise" (Keeney, 1996). Alternatives should be focused at fulfilling the demands of achieving the specified values and should be focused on the generation of a set of the most promising ones. A possible way of discovering new objectives is to think of what alternatives would be available if that particular objective was the only objective on the list, and then taking permutations of objectives and asking the same question. Alternatives should then be combined into single alternatives if possible. Means objectives should also be used for the same reason. At the end of the entire decision process, it is helpful to think if any new alternatives can be generated after the analyst's state of knowledge has been fully updated.

Keeney admits,

> It may initially be difficult to articulate, review, and revise your objectives. You may get the feeling that you are not 'solving' your decision problems when you are just thinking about objectives. You may feel it is merely a philosophical exercise to articulate your values, whereas the decision problems facing you are real. But whether or not you label thinking about your values as an exercise, the results can help with any of the real decisions that you make. One good decision opportunity can repay you for a lot of 'philosophical' thinking" (Keeney, 1996).

In this way, VFT is all about articulating values logically in order to understand both the decision opportunities and additional alternatives that can be developed in cases that without this structure, it would be difficult or impossible to uncover.

A systematic representation of the Value-Focused Thinking process is seen in Figure 18. This chart is comprised of ten different steps and two main subsections. The first step is the identification of the decision problem after careful study and deliberation with all of the Decision Makers and Stakeholders. From this, Step 2 is to create the value hierarchy to understand which objectives are means objectives and which are fundamental objectives, and how the values are related. From this hierarchy, measurements of the values must be decided upon in order to enumerate the fulfillment of the objectives. Step 4 is to create functions based on SME and DM input to understand which thresholds of the fulfillment of the measures should be weighted higher and whether these functions should be categorical or continuous functions. The hierarchy is then weighted in Step 5 using either local or global weights within the total hierarchy or only within the values, weighting each of the measurements against each other one at a time. These five steps complete the first major phase of the process, as now alternatives can be generated using the updated situational awareness that has occurred from logically eliciting the decision formulation.

As discussed previously, alternatives can be uncovered at each step along the way and should be always sought out in a parallel process as more knowledge is created. At this point, the alternatives that are currently in our stead can be scored using the hierarchy, which constitutes the completion of the value model. However, after this stage, Step 8 is to complete Deterministic Analysis in order to weigh each alternative against each specific measure, which can help us find bounds on which alternatives would be useful or chosen under certain situations, and which would be totally out of the question due to the alternatives that are better. Sensitivity Analysis is done in Step 9 in order to change the parameters and discover if there would be any change in the conclusions under different assumptions or desires. Finally, Step 10 is the communication of the Conclusions and Recommendations for the decision situation. From here, the whole process could be iterated if the selected alternative leads to even more decision opportunities. The main point is that the analyst's work is never done, and they must be vigilant and adaptable to new and improved alternatives to make more refined decisions.

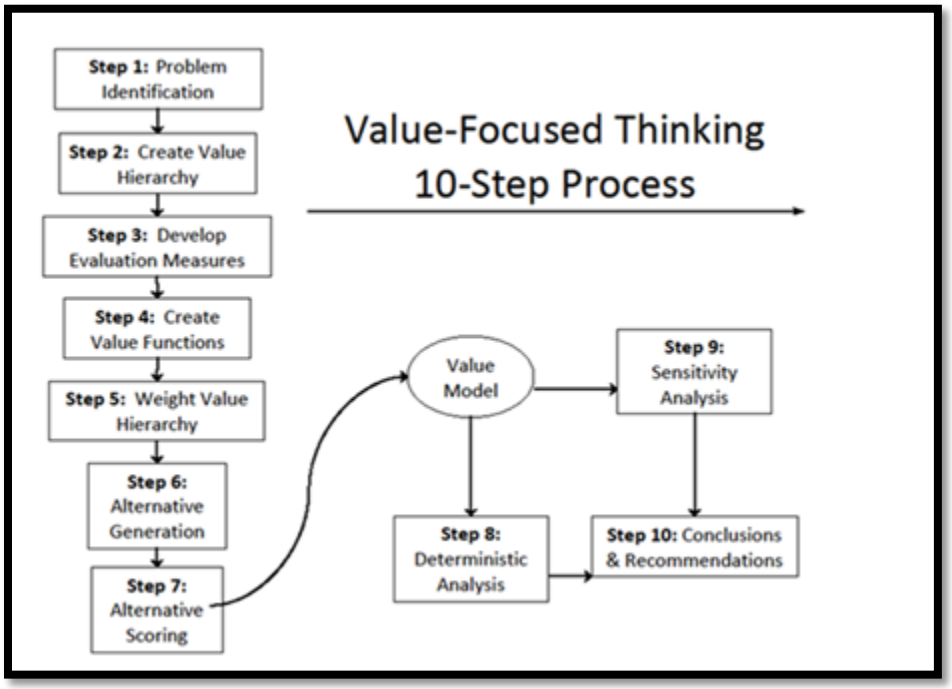


Figure 18. - VFT 10-Step Process (Shoviak, 2001)

An application of the Value Focused Thinking process was utilized by Major Brian Bassham, PhD in 2006 in order to assess the development of Automatic Target Recognition (ATR). Two separate perspectives were studied during his assessment, including the Evaluator's and the Warfighter's. Bassham explains,

> The method involves the development of a two-pronged decision analysis model that maps ATR MOPs (Measures of Performance) into values. This is a direct mapping for the Evaluator. However, the Warfighter thinks more in terms of MOEs (Measures of Effectiveness). To incorporate the Warfighter perspective, a combat model, using a notional, unclassified scenario, was exercised in a designed experiment to produce a response surface that could serve a surrogate and intermediate mapping from MOP to MOE (Bassham, 2006).

This methodology is a unique combination of the VFT process in two different dimensions, the Warfighters and the Evaluators. It is seen in Figure 19. The MOPs in this case constitute the values that would be used in a VFT hierarchy.

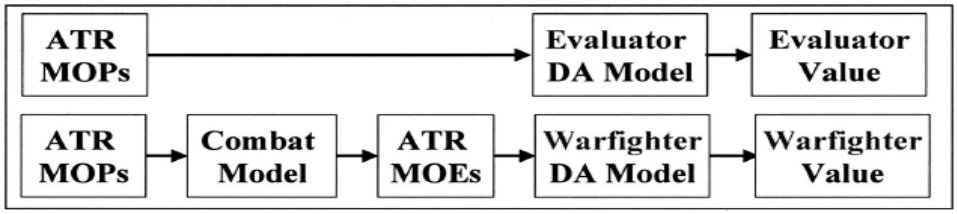


Figure 19. - VFT Process for Evaluators and Warfighters (Bassham, 2006)

The type of decision opportunity framework was slightly different than Keeney's ten-step process, but it had the same elements. As seen in Figure 20, the steps used included first identifying the problem up front, then the objectives and alternatives, and then the next step combined the steps of developing measures, creating value functions, and weighting the value hierarchy in the previous process. This step is considered a decomposition and modeling of the problem in terms of the structure, uncertainties, and preferences inherent in the model. This would be reflected in the previous framework by the weights associated to the values and value functions that are associated with the measurements. From here, the best alternative is chosen,

and then reiterative sensitivity analysis is accomplished to ensure the robustness of the choice of alternative. Finally, after sufficient analysis has been accomplished, the final alternative is chosen and implemented.

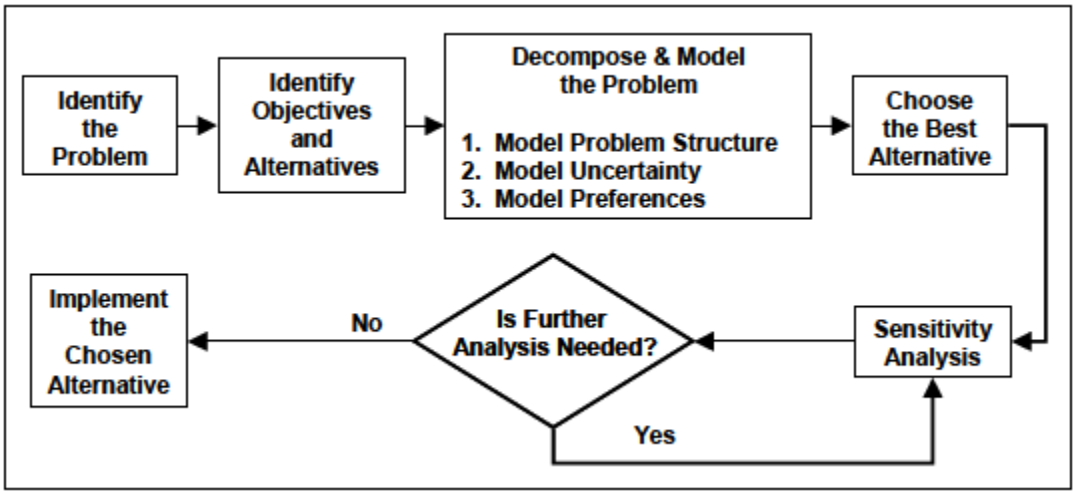


Figure 20. - Bassham's VFT Methodology (Bassham, 2006)

The Hierarchy is seen below in figure 21. The top level values include robustness, classification ability, employment concept, declaration ability, cost, self-assessment accuracy, and overall detection performance. Some ideas from this methodology were used to accomplish the research in this thesis. Some however, are specific to the context of a broader range of ATR than is studied herein.

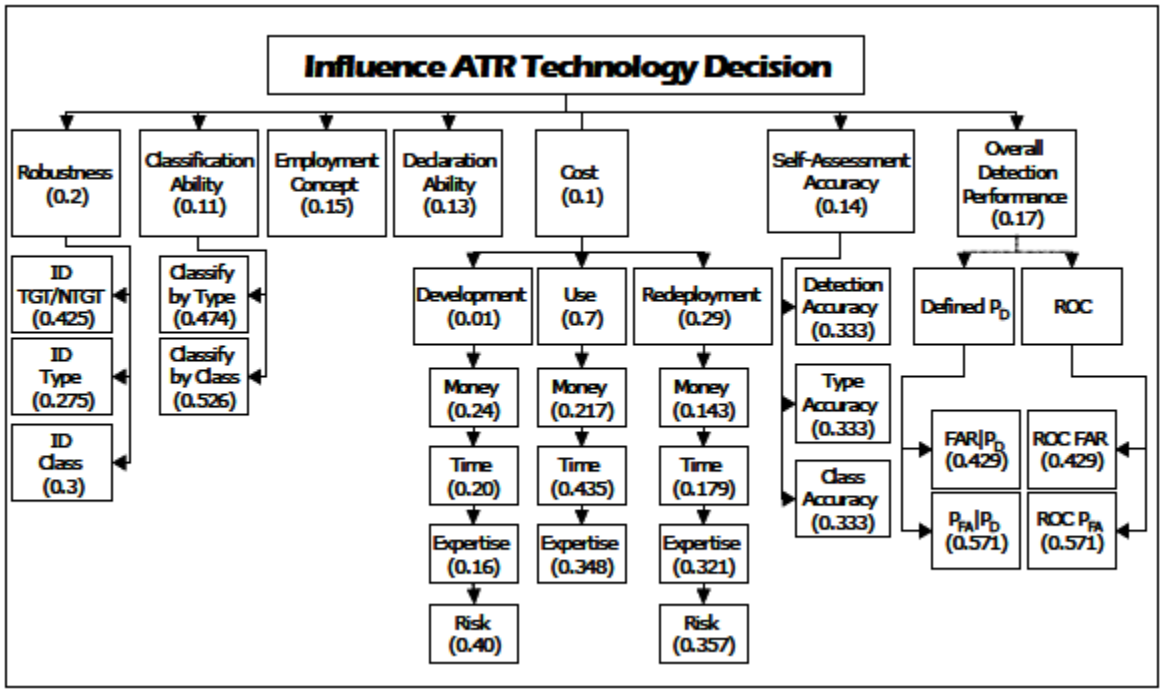


Figure 21. - ATR Value Hierarchy (Bassham, 2006)

The weights that were chosen and implemented as a result of Step 5 of the VFT hierarchy are seen in Table 4. The same delineation of weight and ranking is done here as was done in Keeney's article. This helped discover which objectives were initially under or over prioritized. This same assessment of ranking is done in this research effort and is one of the main cruxes of the advantages of VFT.

Table 4. - Evaluators MOP's (Bassham, 2006)

Objective	MOP	Total Possible Weight	Rank
Robustness	%Δ P_D (TGT/NTGT)	0.0850	4
	%Δ P_{ID} (Type)	0.0550	8
	%Δ P_{CC} (Class)	0.0600	6
Detection Performance	$FAR\|P_D$	0.0729	5
	$P_{FA}\|P_D$	0.0971	3
Employment Concept	Employment Rating	0.1500	1
Declaration Ability	P_{DEC}	0.1300	2
Classification Ability	P_{ID}	0.0521	9
	P_{CC}	0.0579	7
Cost	Development Money	0.0002	21-23
	Development Time	0.0002	21-23
	Development Expertise	0.0002	21-23
	Development Risk	0.0004	20
	Redeployment Money	0.0041	19
	Redeployment Time	0.0052	18
	Redeployment Expertise	0.0093	17
	Redeployment Risk	0.0104	16
	Use Money	0.0152	15
	Use Time	0.0305	13
	Use Expertise	0.0244	14
Self-Assessment Accuracy	$E_{S\text{-}PD}$	0.0466	10-12
	$E_{S\text{-}PCC}$	0.0466	10-12
	$E_{S\text{-}PID}$	0.0466	10-12

The same type of methodology for creating weights and values structured within objectives was utilized for the Warfighter's perspective using the Measures of Effectiveness (MOEs). This is seen in Table 5. Ranks and values were elicited and compared for each MOE. Bassham states,

> A major complication in the decision-making process is the fact that picking a 'best' system based on MOPs does not necessarily lead to superior operational performance. In an operational environment the system is characterized by measures of effectiveness (MOEs), which are qualitative and quantitative measures of how well tasks are performed. The eventual end-user, the Warfighter, cares only about the benefit ATR technology offers in battle. Thus, selecting an ATR CS based upon superior MOEs would be of great interest to the Warfighter (Bassham, 2006).

The final results of this study concluded that the evaluator and warfighter would pick two different optimum alternatives to put in place. This means that there should be some type of satisficing or mediation done to ensure that both parties are confident with the chosen option.

Table 5. - Warfighter's MOE's (Bassham, 2006)

Objective	MOE	Total Possible Value	Rank
Minimize Hostile Weapons	% of Bombs Left	0.0102	14
	% of Mass Destruction Left	0.0596	5
	% of CMs & S/S Left	0.0357	8
	% of S/A & A/A Left	0.0513	6
Minimize Hostile Warfighting Systems	% of Systems Left	0.2149	2
	% of Personnel Left	0.0682	4
	% of C2 Left	0.2977	1
Minimize 'Bad Press'	Length of Battle	0.0124	13
	# of Civilians Killed	0.0241	11
	# of Civilian Structures Destroyed	0.0134	12
	# of Fratricide Incidents	0.0877	3
Maximize Friendly Weapons Remaining	% of Systems Left	0.0279	9-10
	% of Personnel Left	0.0457	7
	% of C2 Left	0.0279	9-10
Maximize Friendly Warfighting Systems Remaining	% of Dumb Bombs Left	0.0032	18
	% of Precision Bombs Left	0.0073	15
	% of CMs & S/S Left	0.0064	16-17
	% of S/A & A/A Left	0.0064	16-17

The three different objectives that were developed using the Warfighter's viewpoint included Maximizing the Effect on the Enemy, Minimizing Unintended Consequences, and Minimizing the Effect on Allies. The effect on the enemy was weighted the highest, due to the correlation of this objective with the success of the mission. This objective was broken down into Minimizing Expendables Remaining and Minimizing Warfighting Systems Remaining. These two means objectives represent a combination of the likelihood that the enemy will be crippled by exercising the mission. Minimizing Bad Press was itself a primary objective that was weighed by four different measurements that dealt with quantities of events that would be detrimental to the viewpoint of the military in the eyes of the general public. On the other side of the coin, the third primary objective, Minimizing Effects on the Allies, was split into the same types of means objectives as the Enemy Effect objective, although in this case, it is maximizing the warfighting systems and expendables remaining. A combat model was created to simulate the effects of the ATR technology in an operational environment. This breakdown is shown in Figure 22.

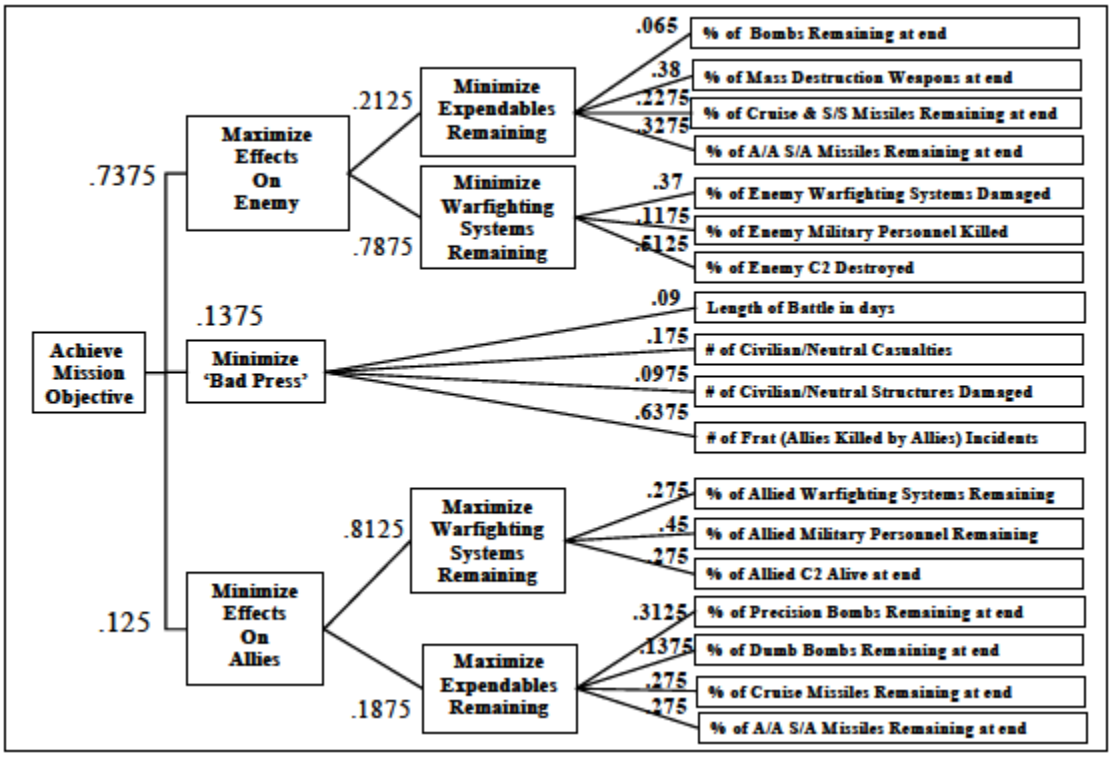


Figure 22. - Warfighter Value Hierarchy (Bassham, 2006)

This example highlighted some of the challenges and issues that could be discovered in a real-world operational situation. Carrying out the analysis to completion helped elicit the challenges that would occur in making a structured decision, but it also helped to understand which alternatives should not be considered in future situations under the same assumptions. Practice makes perfect, and each time a decision situation is analyzed, the analyst becomes more capable of understanding the nuances that arise and this prepares them for more complicated scenarios.

Bias/Variance Dilemma

The success and quality of a particular classifier can be analyzed using the bias-variance decomposition of the classification error. When assessing the difference between the estimated density of the class label frequency within a set of data and the true density of that set, a useful

statistic to use is the Mean Squared Error (MSE) between these two densities (Dougherty, 2013). This MSE can be composed of a combination of error due to the bias inherent in the classifier and the variance. When training classifiers, if the classifier is not flexible enough too few parameters) to estimate near the expected values for the class labels, then the classifier will exhibit high bias. If the classifier becomes too flexible (too many parameters), the classifier is known to over-fit the predictions towards the instance of the training data set. If this occurs, the classifier is said to exhibit high variance. This would mean that it is predicting different class labels when it is exposed to different training sets. Many supervised learning classifiers can be tuned in terms of this trade-off automatically or by containing a parameter than can be manipulated by an analyst.

An illustration of the ideas of bias and variance are found in Figure 23, where dart boards are used as examples (Fortmann-Roe, 2014). In the upper left board, the darts have been thrown both near the center of the target, exhibiting low bias, and with a high level of precision, as they are clumped together. In the upper right board, the darts have been thrown near the center of the target, but with less precision, as there is higher variability in their locations. The board on the bottom left has a high precision value, as all of the darts are packed together, while they miss their mark, exhibiting high bias. Finally on the bottom right, the darts are neither packed tightly together nor thrown near the center of the target, meaning that the process is exhibiting both high variance and bias.

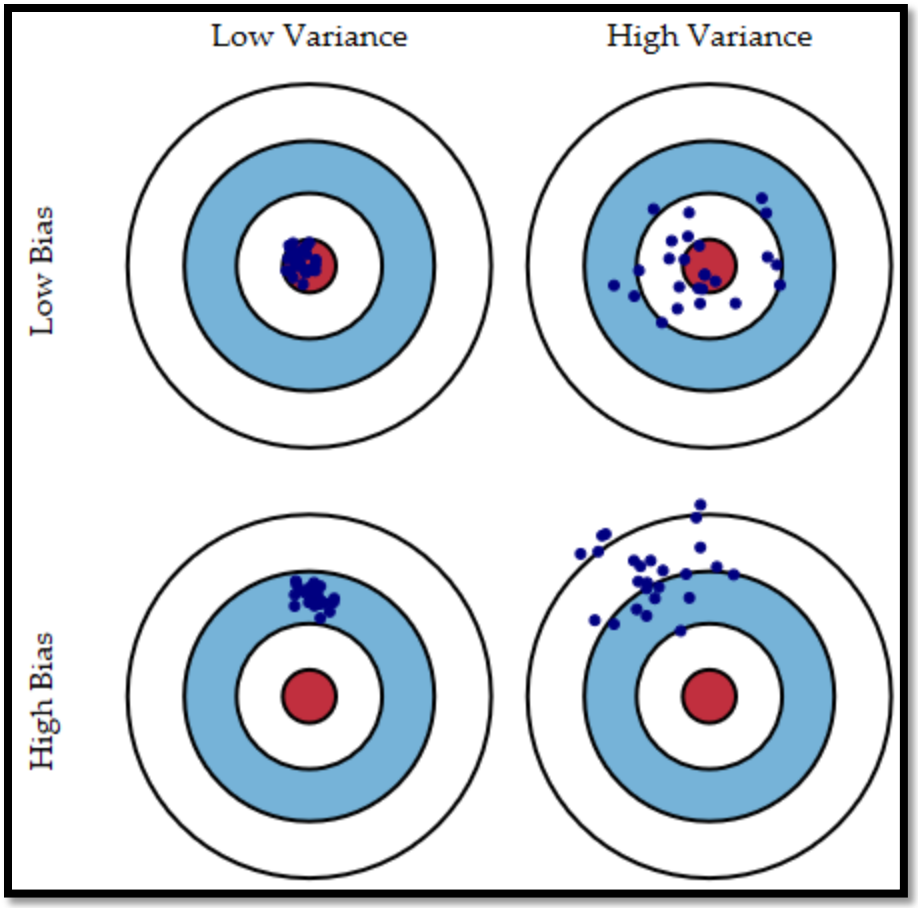


Figure 23. - Bias and Variance Comparisons (Fortmann-Roe, 2014)

Figure 24 depicts the phenomenology of the relation between bias and variance and the training and test sets that are used in the classification approach. As the model complexity increases and more flexibility is built in the model with additional parameters, the bias, or closeness of the model predicting towards some target representation of truth, decreases, but the variation of the classifier's modeling ability to additional sets of data, the variance, increases. The prediction error is comprised of the bias and variance components and it decreases in training, as seen in the blue curve, as the model complexity increases, since the model is being fit to the training sample. However, the red curve, which represents the prediction error across model complexity for test samples, would simultaneously increase due to this increase in

variance. It is important to find the point where variance is tempered for the test sample. At this point, the model has been adequately trained and tested and can move on to the validation stage.

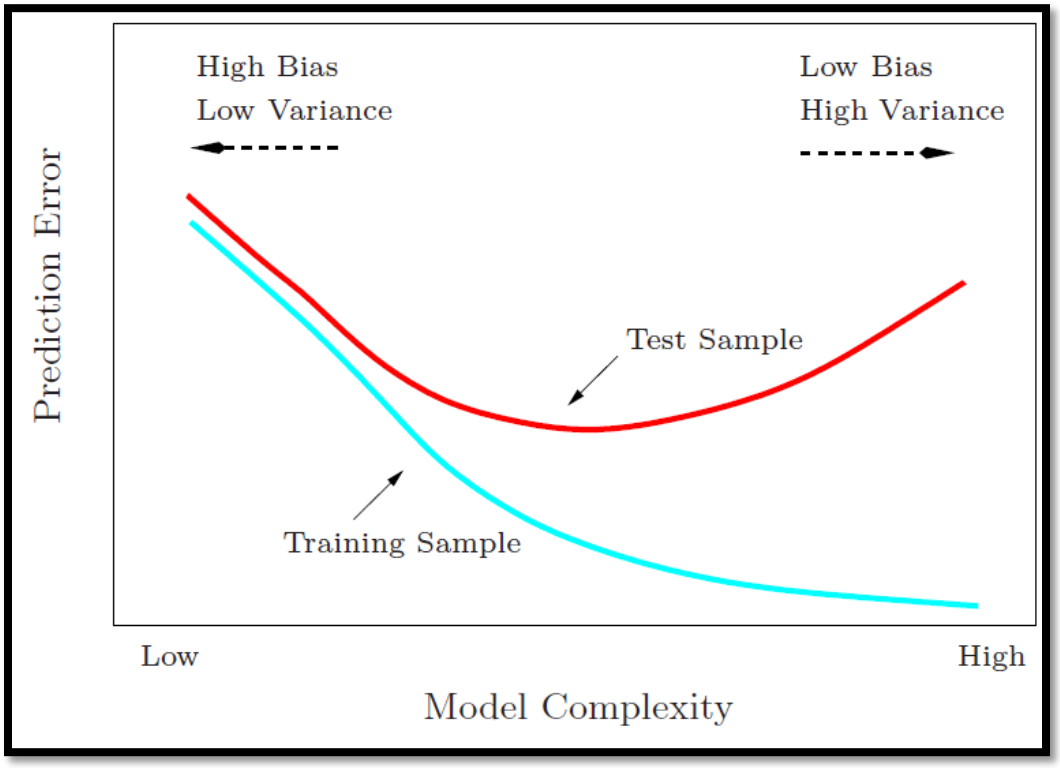


Figure 24. - Bias and Variance per Model Complexity (Hastie et al., 2009)

Dougherty explains the concept and importance of over-fitting,

> The No Free Lunch Theorem throws into question our preference for avoiding over-fitting and choosing the simplest classifiers with fewer features and parameters. In the former case, there are indeed problems for which avoiding over-fitting actually leads to worse performance. It is not over-fitting *per se* that causes poor performance; it is rather the mismatch of the algorithm (in use) to the specific algorithm (describing reality). As for simple classifiers (in line with Occam's razor), our bias towards simple solutions may have an evolutionary basis, i.e., there is a strong selection pressure for simple schemes which require fewer neurons and less computational time (Dougherty, 2013).

Therefore, the idea of over-fitting stems from assuming that the algorithm that has been developed is in actuality the correct formulation for a generalized version of reality, when better algorithms are actually more suitable.

Figure 25 displays the relationship between the number of parameters in the model to the error decomposition components of the loss function, $Bias^2$ and $Variance$. As the number of

parameters increases, the model becomes more flexible, and the model fits to the data better. This means the $Bias^2$ value decreases. As the number of parameters increases, the data also over-fits to the specific instance of data, including the noise that is inherent within that dataset. Therefore, the variance increases as the number of parameters and thus the flexibility increases.

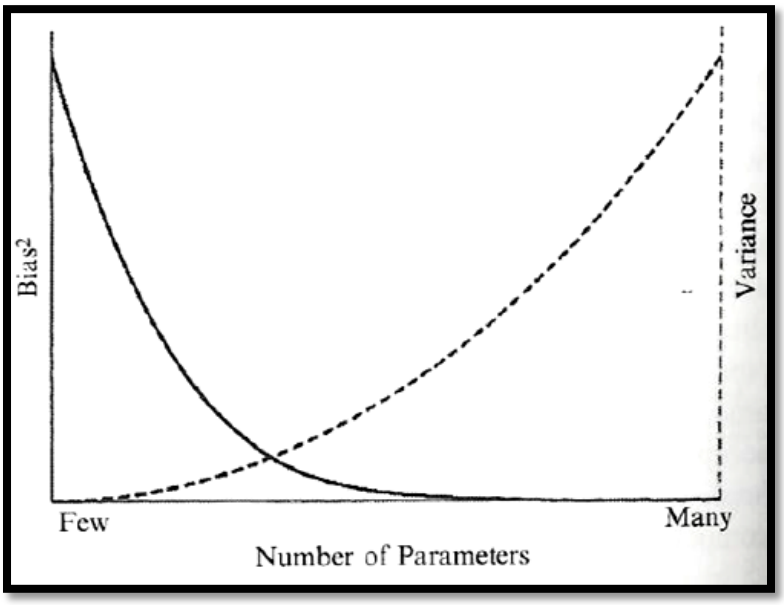


Figure 25. - Bias and Variance per Number of Parameters (Dougherty, 2013)

The MSE is defined as (Dougherty, 2013):

$$MSE(x) = E\{(\bar{x} - x)^2\} \tag{12}$$

where

x is a single parameter, be it a pixel class of interest or some particular estimate of classifier performance.

$\bar{x}$ is the estimate of the parameter of interest.

E is mathematical expectation.

The bias and variance in this case is defined as the following (Dougherty, 2013):

$$B(x) = E\{(\bar{x} - x)\} \tag{13}$$

$$V(x) = E\{(\bar{x} - E(\bar{x}))^2\} \tag{14}$$

The MSE value is decomposed in terms of bias and variance as (Dougherty, 2013):

$$MSE(x) = B^2(x) + V(x) \tag{15}$$

A useful example is found within the context of the K-Nearest Neighbors classification algorithm. In K-Nearest Neighbors, the class of a point of interest is predicted by choosing an some number of "neighbors" of the point using some choice of distance value. A voting method is used to determine the class of the point of interest using these nearest points. If too many points are chosen, the classifier is said to exhibit bias, as the estimated class will not usually correspond with the actual class, since the classifier is not flexible enough. However, if too few points are chosen, the classifier is said to exhibit high variance, as the predictions will be fit too tightly, and any new data sets that are used will not result in quality estimates. When there is high variance, the classifier is thought of as fitting the estimates to the noise of the particular data set.

The key to optimizing the classifier is to find an amount of flexibility resulting in a reduction of bias and variance to an acceptable level. Definitions for the bias and variance in classification tasks are a still open research topic. There have been multiple proposals for the definition and calculation of these values, which will be seen in following paragraphs.

The state of nature at a point *x* in the overall feature space is a random variable $\omega \in \Omega$, where Ω is the overall set of possible class labels, and ω is the class label that is chosen. The classifier is used to determine the state of reality by picking the best class label it possibly can from Ω (Kuncheva, 2004). The true values of the posterior probabilities for the multiple classes

are $P(\omega_i|x)$, and the probabilities across all the possible classifiers that could have been chosen is $P_D(\omega_i|x)$, which represents the guessed state of nature for that classifier (Kuncheva, 2004). P_D is the probability for a specific training data set D. This represents the probability that a randomly chosen classifier will assign the particular class label for the point of interest. $\Sigma_i P(\omega_i|x) = 1$ and $\Sigma_i P_D(\omega_i|x) = 1$ (Kuncheva, 2004).

Dougherty states that "The bias of an estimate is the systematic error incurred in the estimation; the variance of an estimate is the random error incurred in the estimation" (Dougherty, 2013). The idea of bias for any estimate is the averaged difference between the true and predicted values. In this particular case, this means that bias represents the difference between the true correct probability of choosing the correct class for the point in the feature space and the estimated distribution, $P_D(\omega_i|x)$.

The idea behind variance is thought of as a measurement of the precision of the classifier in predicting the class of the point, independent of what is actually going on. Typically for a random variable, the measure of variability is its variance, but this is not the case for categorical variables such as the class label of the point. Entropy is often used as an estimator for a categorical variable, which is formulated as the following:

$$H = -\sum_{\omega_i} P_D(\omega_i|x) \log P_D(\omega_i|x) \tag{16}$$

In this case, $H = 0$ occurs when no variability is present, while $H = \log c$, where c represents the number of classes in the data, occurs when the variability is the highest, and each class label has the same probability of being chosen (Kuncheva, 2004). Gini Index is sometimes chosen as the variance, which is represented as the following:

$$G = \sum_{\omega_i} 1 - P_D(\omega_i|x)^2 \tag{17}$$

Noise is a measure of variability inherent in reality that does not depend on the choice or quality of the classifier being used. *H* and *G* are often used for noise as well (Kuncheva, 2004).

According to Kuncheva, Kohavi and Wolpert define the bias, variance, and noise as the following (Kuncheva, 2004):

$$bias = \frac{1}{2} * \sum_{\omega_i} \left(P(\omega_i|x) - P_D(\omega_i|x)\right)^2 \tag{18}$$

This bias represents the difference between the true distribution of the particular class and the guessed one for the specific data set, and this difference is squared and added for each specific class. Each guessed distribution and probability can be generated from a bootstrap sample (Kuncheva, 2004).

$$\text{Variance} = \frac{1}{2}(\sum_{\omega_i} 1 - P_D(\omega_i|x)^2) \tag{19}$$

The variance can be seen as the change in the best estimate the classifier makes for each class distribution regardless of what the true distributions are. The variance that is used here is the Gini index. Each guessed probability of the class distributions will be generated from individual bootstrap samples (Kuncheva, 2004).

$$noise = \frac{1}{2}(\sum_{\omega_i} 1 - P(\omega_i|x)^2) \tag{20}$$

This noise value can be interpreted as the variability that is inherent to the process regardless to the classifier used. The decomposition of the error term into bias, variance, and noise are different for each of these formulations. For Kohavi and Wolpert, the following is the breakdown (Kuncheva, 2004):

$$P(error|x) = 1 - \sum_{\omega_i} P(\omega_i|x)P_D\,(\omega_i|x) + \frac{1}{2} * \sum_{\omega_i} P\,(\omega_i|x)^2 + \frac{1}{2} * \sum_{\omega_i} P_D\,(\omega_i|x)^2$$

$$-\frac{1}{2} * \sum_{\omega_i} P\,(\omega_i|x)^2 - \frac{1}{2} * \sum_{\omega_i} P_D\,(\omega_i|x)^2 \tag{21}$$

$$= \frac{1}{2} * \left(\sum_{\omega_i} (P(\omega_i|x) - P_D(\,\omega_i|x))^2 \right) + \frac{1}{2} * (-\sum_{\omega_i} P_D(\omega_i|x)^2) + \frac{1}{2} * (1 - \sum_{\omega_i} P(\omega_i|x)^2)$$

$$= bias + variance + noise$$

According to Kuncheva, Breiman defines a noise, bias, and spread term for bias and variance decomposition. The noise term is the same as the Bayes error for that x (Kuncheva, 2004).

$$noise = 1 - P(\omega^*|x) \tag{22}$$

$$bias = (P(\omega^*|x) - P(\hat{\omega^*}|x))P_D(\hat{\omega^*}|x) \tag{23}$$

where

ω^* is the most probable class of the specific x vector

$\hat{\omega^*}$ is the highest likelihood output for the specific x vector from a specific classifier

This bias is always nonnegative due to the maximization of $P(\omega_i|x)$ by ω^*. The variance term is Breiman's formulation is labeled as a spread term, which shows how the distribution that is guessed by the classifier changes between class labels outside of the ω^* and $\hat{\omega^*}$ labels (Kuncheva, 2004).

$$spread = \sum_{\omega_i \neq \omega*} \big(P(\omega^*|x) - P(\omega_i|x)\big)P_D(\omega_i|x) \tag{24}$$

Breiman's error decomposition is seen in Equation 25 (Kuncheva, 2004):

$$P(error|x) = 1 - \sum_{\omega_i} P(\omega_i|x)P_D\,(\omega_i|x) = 1 - P(\omega^*|x) + P(\omega^*|x)\sum_{\omega_i} P_D(\omega_i|x)$$

$$- \sum_{\omega_i} P(\omega_i|x)P_D(\omega_i|x)$$

$$= 1 - P(\omega^*|x) + \sum_{\omega_i}\big(P(\omega^*|x) - P(\omega_i|x)\big)\,P_D(\omega_i|x) \tag{25}$$

$$= 1 - P(\omega^*|x) + \Big(P(\omega^*|x) - P\big(\omega^{\hat{*}}\big|x\big)\Big)\,P_D\big(\omega^{\hat{*}}\big|x\big)$$

$$+ \sum_{\omega_i}\big(P(\omega^*|x) - P(\omega_i|x)\big)\,P_D(\omega_i|x)$$

$$= noise + bias + spread$$

Domingos takes a different approach to defining bias, variance, and noise, in that he attempts to develop a uniform definition across all loss functions.

$l(T(x), D(x))$ is the loss for x that occurs when a randomly selected classifier D is applied to a vector, x, and where $T(x)$ is the true label at that specific x vector and $D(x)$ is the guessed label. The bias for the specific x value is:

$$bias = l(\omega^*, \omega^{\hat{*}}) \tag{26}$$

The bias is independent of the classifier that is being used to determine class label and is only dependent on the most often guessed label (using resampling), which is considered the majority label, and designated by $\omega^{\hat{*}}$ and the optimal class label for the particular *x*, which is ω^*. This results in bias that is either 0 or 1 depending upon if the label is matched to the optimum class (Kuncheva, 2004).

$$variance = \varepsilon_D(l\left(\omega^{\hat{*}}, D(x)\right)) \tag{27}$$

For the 0/1 loss function, the variance is

$$variance = \sum_{\omega_i \neq \hat{\omega^*}} P_D(\omega_i|x) = 1 - P_D(\hat{\omega^*}|x) \tag{28}$$

$$noise = \varepsilon_T(l(T(x), \omega^*)) \tag{29}$$

The noise stems from the data set distributions that are being analyzed and is independent of specific classifiers. For the 0/1 loss function, the noise equation works out to:

$$noise = 1 - P(\omega^*|x) \tag{30}$$

The best model will result in the posterior probabilities that correspond to the optimal class label probabilities, which means that $P(\omega_i|x) = P(\omega^*|x)$. Kuncheva then discusses the philosophy of each member of the bias and variance decomposition for particular loss functions.

"Then the bias measures how far the majority prediction is from the optimal prediction, the variance shows the variability of the predicted label about the majority prediction, and the noise tells us how far the optimal prediction is from the truth (Bayes error)" (Kuncheva, 2004).

The error decomposition for Domingos is different depending on the loss function that is employed.

$$P(error|x) = c_1 * noise + bias + c_2 * variance \tag{31}$$

c_1, c_2 are either constants or expressions that depend on the utilized loss function. For zero-one loss, expressions are used for these variables that depend on the bias, variance, and noise functions.

$$P(error|x) = P(\omega_1|x)P_D(\omega_2|x) + P(\omega_2|x)P_D(\omega_1|x) \tag{32}$$

When the example vector is unbiased, $bias = 0$

$$P(error|x) = (1 - 2 * noise) * variance + noise \tag{33}$$

When the x is unbiased, the probability of error decreases when the variance decreases. This makes sense, as variance dominates the noise, and focus should be on training the classifier to decrease this variance component.

When the example vector is biased, $bias = 1$, the decomposition is the following

$$P(error|x) = bias + (2 * noise - 1) * variance - noise \quad (34)$$

Therefore, against intuition, for these biased examples, increasing the variance actually decreases the amount of error. This may explain why using a large ensemble of biased classifiers will decrease the total error of classification (Kuncheva, 2004).

Kuncheva describes the relationship between bias and variance for certain situations,

> All decompositions of the error are aimed at studying the structure of the error for different classifier models and ensembles of classifiers. Suppose that we build our random classifier *D* using different data sets drawn from the distribution of the problem. It is natural to expect that simple classifiers such as the linear discriminant classifier will have high bias (deviation from the optimal model) and low variance. Conversely, flexible classifiers such as neural networks and decision trees will vary significantly from data set to data set because they will try to fit the particular realization of the data as close as possible. This means that they will have high variance but their bias will be low (Kuncheva, 2004).

The fluidity of the bias and variance quantities changes as parameter estimates are changed. This can be used to the advantage of the analyst, if the correct parameter manipulation is done. However, sometimes, the only choice is making a decision on a trade-off between bias and variance.

> If the classifier has a parameter that we can tune, then making the classifier more coarse and robust will diminish its sensitivity, therefore will decrease the variance but might increase the bias. Sometimes varying a classifier reduces both bias and variance, thereby giving a smaller error altogether for certain data sets. This is seen in the *k*-nearest neighbor classifier. For tree classifiers, the control parameter may be the depth of the tree or the constant used in prepruning. Typically, heavily pruned trees will have smaller variance and larger bias than trees fully grown to classify correctly all training samples (Kuncheva, 2004).

Friedman's Formulation

Bias and Variance for Classification

Friedman was the first statistician to attempt to separate the bias and variance decomposition from the regression case to the classification case. In the classification case, the following is true for a zero/one loss function (Friedman, 1997) (Duda et al., 2001).

The following is Target/Discriminant function (Duda et al., 2001):

$$F(x) = \Pr[y = 1|x] = 1 - \Pr\,[y = 0|x] \tag{35}$$

The discriminant function is thus:

$$y = F(x) + \epsilon \tag{36}$$

$$Var[\epsilon|x] = F(x)(1 - F(x)) \tag{37}$$

The target function is thus:

$$F(x) = \varepsilon[y|x] \tag{38}$$

Mean Square Error is minimized (Equation 39) (Duda et al., 2001):

$$\varepsilon_D[(g(x; D) - y)^2] \tag{39}$$

If equal priors are assumed:

$$P(\omega_1) = P(\omega_2) = 0.5 \tag{40}$$

then the Bayes discriminant, y_B, equals $\frac{1}{2}$. The Bayes decision boundary will be the locus defined by $F(x) = \frac{1}{2}$ (Duda et al., 2001). The classification error rate (averaged over each specific x vector), Pr $[g(x; D) = y]$, will result in the lowest error if it corresponds with the Bayes decision boundary,

$$\Pr[g(x; D) = y] = \Pr[y_B(x) \neq y] = \min[F(x), 1 - F(x)] \tag{41}$$

(Friedman, 1997) (Duda et al., 2001).

If not, then the prediction yields an error that is increased, seen here (Friedman, 1997) (Duda et al., 2001):

$$\Pr[g(x;D)] = \max[F(x), 1 - F(x)] = |2F(x) - 1| + \Pr[y_B(x) = y] \quad (42)$$

This error is averaged over all of the data sets to derive the following (Friedman, 1997) (Duda et al., 2001):

$$\Pr[g(x;D) \neq y] = |2F(x) - 1| \Pr[g(x;D) \neq y_B] + \Pr[y_B \neq y] \quad (43)$$

Duda et al. states, "The classification error rate is linearly proportional to $\Pr[g(x;D) \neq y_B]$, which is the "Boundary Error", since it represents the incorrect estimation of the optimal Bayes boundary" (Duda et al., 2001) (Friedman, 1997). Since each training set contains its own noise, the boundary error will change with the probability density of obtaining a specific discriminant, which is denoted as $p(g(x;D))$ (Duda et al., 2001). This boundary error is the captured by the area of the tail of $p(g(x;D))$ on the other side of the Bayes discriminant value ½ (Duda et al., 2001). In Figure 26, if the class from the normal population was being predicted, but the class was truthfully the abnormal population, the boundary error would be represented by the tail of the abnormal population, which is the area designated by b, which is the side opposite of the Bayes optimum.

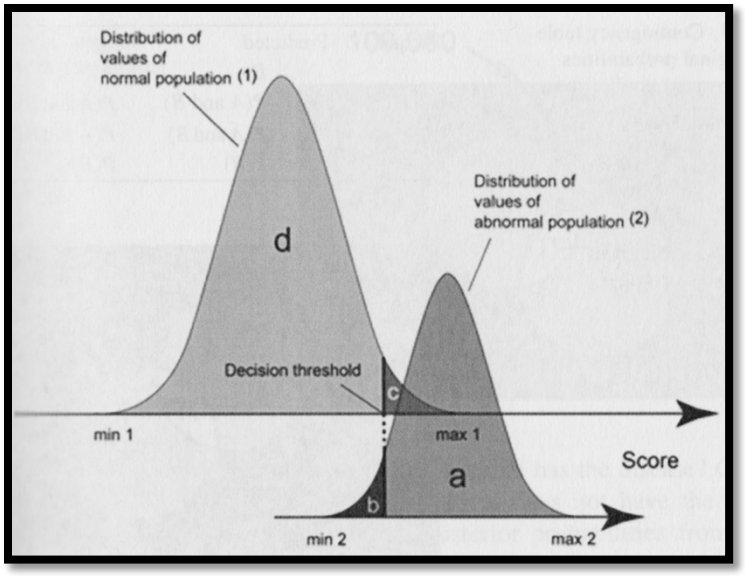


Figure 26. - Representation of Boundary Bias (Dougherty, 2013)

The formula for the area under this tail is given by the following formulation (Duda et al., 2001):

$$\Pr[g(x;D) \neq y_b] = \begin{cases} \int_{1/2}^{\infty} p\big(g(x;D)\big)dg & if\ F(x) < 1/2 \\ \int_{-\infty}^{1/2} p\big(g(x;D)\big)dg & if\ F(x) \geq 1/2 \end{cases} \tag{44}$$

Assuming that $p(g(x;D))$ is Gaussian, the following bias, variance decomposition can be made (Duda et al., 2001):

$$\begin{aligned} \Pr[g(x;D) \neq y_B] &= \Phi\left[Sgn\left[F(x) - \frac{1}{2}\right]\frac{\varepsilon_D[g(x;D)] - \frac{1}{2}}{\sqrt{Var[g(x;D)]}}\right] \\ &= \Phi[Sgn\left[F(x) - \frac{1}{2}\right]\left[\varepsilon_D[g(x;D)] - \frac{1}{2}\right]Var[g(x;D)]^{-\frac{1}{2}}] \end{aligned} \tag{45}$$

In this case, the boundary bias is represented by $Sgn\left[F(x)-\frac{1}{2}\right]\left[\varepsilon_D[g(x;D)]-\frac{1}{2}\right]$ and the variance is represented by $Var[g(x;D)]^{-\frac{1}{2}}$

The function designated by phi is the following (Duda et al., 2001):

$$\Phi[t] = 1/\sqrt{2\pi}\int_t^{\infty} e^{-1/2u^2}du = 1/2[1-\text{erf}\left(\frac{t}{\sqrt{2}}\right)] \tag{46}$$

$$\text{erf}(u) = \frac{2}{\sqrt{\pi}}\int_0^u e^{-x^2}dx \tag{47}$$

The erf function that is used here is approximated by the following distributions.

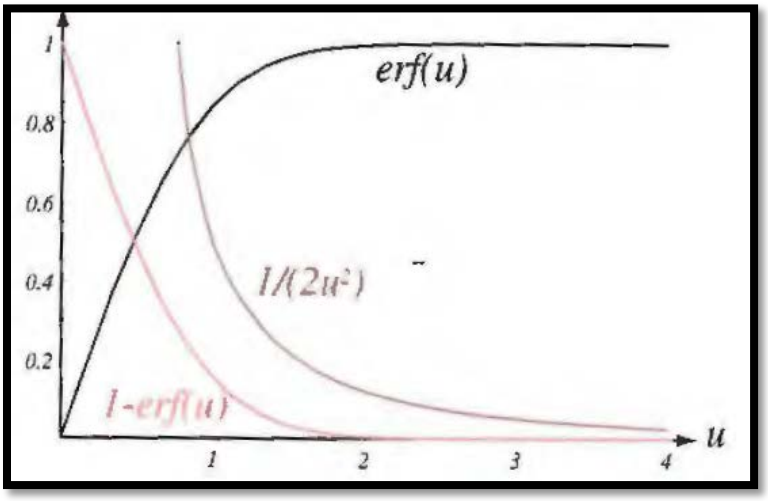


Figure 27. - erf function values (Duda et al., 2001)

Figure 28 shows the bias/variance decomposition in the framework of regression. The rows in the figure represent different training datasets. Moving from left to right, the bias decreases from columns a to b. In column a, the $g(x)$ function is fixed and is a poor estimate of the data, no matter which training set is used. Therefore, no matter what the true function $F(x)$ is, the function will remain fixed for each training set. In the last row, the bias is seen to be large for the first column, as the estimate is very poor. The variance in this case is zero, and the overall error loss function is completely dependent on the bias term. Column b is a slightly better

estimate than column a, even though it is still fixed and has zero variance. Prior knowledge was used to move this function closer to the true distribution of the data and thus the bias, and the overall error, has been decreased. The model in Column C is a cubic model with coefficients that can be trained and changed. The fit to the data is accurate and thus the bias is low. The model in column d is linear but certain parameters, including the intercept and slope have been estimated from the training data. It is not as flexible as the previous model due to the number of parameters, but it is better than the first two models. Therefore the model has a larger bias than the third model and a smaller bias than the first two models. However, due to its flexibility, this model and the third model propagate error through their variance terms. Having prior information, about the system being modeled and the mechanism that generates the data, that can be fed into this flexibility will help decrease both the bias and the variance. Duda et al. explains,

> We can virtually never get zero bias and zero variance; to do so would mean that there is only one learning problem to be solved, in which case the answer is already known. Furthermore, a large amount of training data will yield improved performance so long as the model is sufficiently general to represent the target function (Duda et al., 2001).

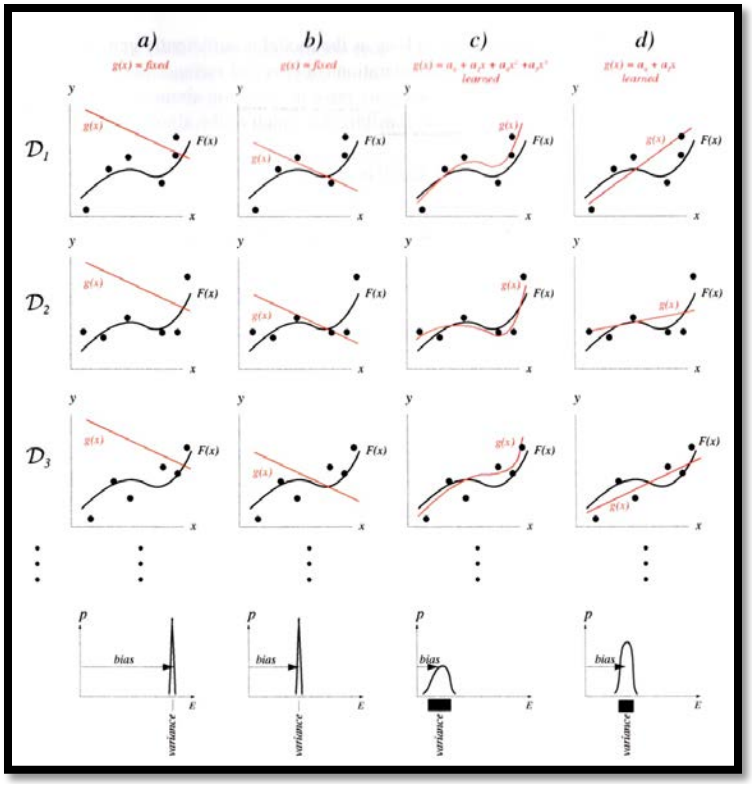


Figure 28. - Regression Bias and Variance (Duda et al., 2001)

In contrast with the regression situation for bias/variance, the classification situation yields a different generalization for the propagation of the two types of error that will be entered in a loss function. In this case, for a two-class problem, samples are drawn from multivariate Gaussian distributions with two different covariance matrices and means. By considering the representation of the covariance matrices, the factor that is being changed across columns, with the left column being the lowest biased distribution representation with off-diagonal covariances, as this full covariance matrix can better estimate the distribution of the classes, while the middle column has zeroed covariances, and the rightmost column has identity covariance matrices, which is the least flexible, and thus highest biased, the relationship between bias and variance in a classification context can be studied. Each row represents instances of the dataset that come from the truth distributions seen on top of the figure. Maximum-likelihood estimation was used

to estimate the parameters and thus separate the classes for a few data points from each class, with the resulting classifiers shown by the dashed lines. Duda et al. explains,

> Notice that most feature points in the high-bias cases retain their classification, regardless of the particular training set (i.e., such models have low variance), whereas the classification of a much larger range of points varies in the low-bias case (i.e. there is high variance). While in general a lower bias comes at the expense of higher variance, the relationship is nonlinear and multiplicative (Duda et al., 2001).

Therefore, the regression situation is not exactly the same as the classification situation for decomposition.

The bottom of the figure shows three density plots that correspond with the different decision boundaries that are developed for many training sets. The gray noisy representation in the leftmost plot shows that there is a high variance in where those decision boundaries are drawn, while for the highest biased situation, the variance is low as the middle of the plot is more dense and black. The average of all of these decision boundaries in the left plot represents the most accurate representation of the true decision boundary, as the bias is low, and this fact is represented by the error histograms seen below. The rightmost plot average boundary would have a larger error, as there is a larger bias from the true boundary, and the error histogram is more peaked than the other two. In order to optimize the bias and variance in a classification context, there must be an adequate amount of data in a training set, which would shrink the amount of error for the given bias, and the number of parameters in the model must be adequate enough to have a good resolution. Duda explains,

> If a model is rich enough to express the optimal decision boundary, its error distribution for the large n case (with decreased variance) will approach a delta function at $E = E_B$, which is the Bayes error. To achieve the desired low generalization error it is more important to have low variance than to have low bias. The only way to get the ideal of zero bias and zero variance is to know the true model ahead of time, in which case no learning was needed anyway (Duda et al., 2001).

When *n* is increased in this case, in order to determine a classifier, more parameters need to be estimated for the model, which would by default decrease the bias. Prior knowledge must be used to find this adequate representation of the model, just like in the regression case.

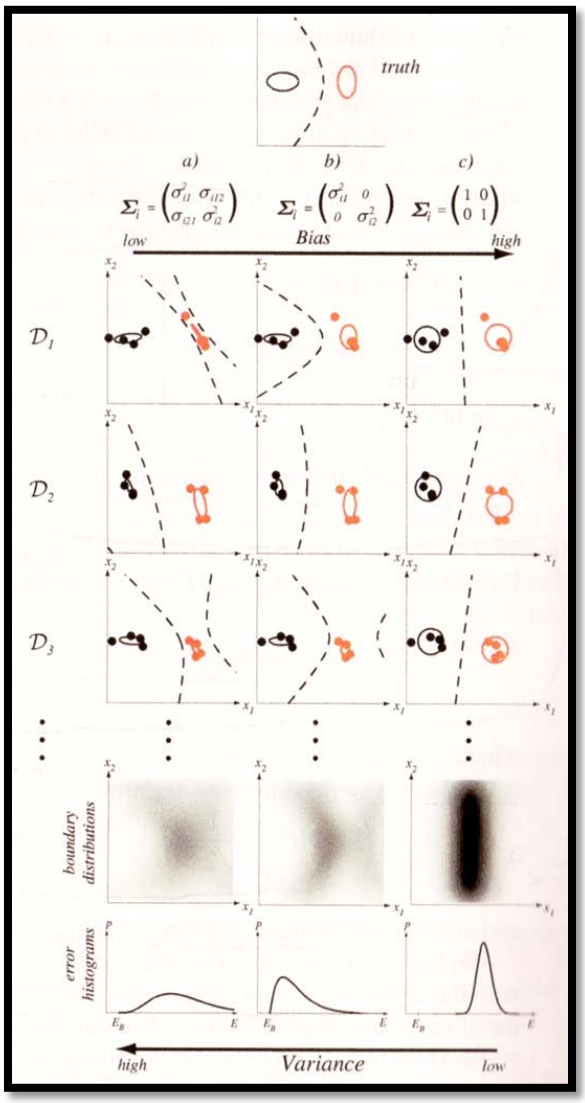


Figure 29. - Classification Bias and Variance (Duda et al., 2001)

III. Methodology

Overview

The main thrust of this research effort is the fusion of different responses of interest with the input of Subject Matter Expertise using the hierarchical framework and ideas of Value Focused Thinking. This fusion of responses will be used to compare algorithms of similar assumptions and of similar computational complexity. While the amounts of responses and the types of responses can be manipulated from study to study, this framework can be used as a baseline methodology that can be used for further research in the realm of HSI data. This will provide a large benefit over the current, disjoint methodology of comparison, which includes comparisons via parametric and nonparametric tests that incrementally exhibit the benefit of the new algorithm or the incremental changes in the algorithms over the previous ones.

The research effort is divided into three main components. The first component is the application and utilization of the Value Focused Thinking Ten-Step Process to the problem of the quality of a hyperspectral anomaly detector in various settings. The second and third components are experiments that include the exercise of the completed hierarchy for the analysis and comparison of different algorithms and imaging data sets. The first experiment is a comparison of three basic supervised algorithms on a series of synthetically permutated images in order to verify the performance of the hierarchical response. The second experiment is focused on using the lessons learned in the first experiment and applying them to hyperspectral image data to understand whether the same results would hold up and the same alternative would be chosen under this different assumption of the level of data.

Value Focused Thinking

The Value-Focused Thinking 10-Step Process was utilized to structure the responses that are of interest in a supervised classification context, and then ultimately in an HSI modeling situation, in order to choose the best alternative from a set of algorithms. This type of analysis allows the analyst to make decisions that are founded on logic and can be repeated for future research or applied to different alternatives. It also is a very transparent process that can be easily communicated and understood by Decision Makers and other analysts.

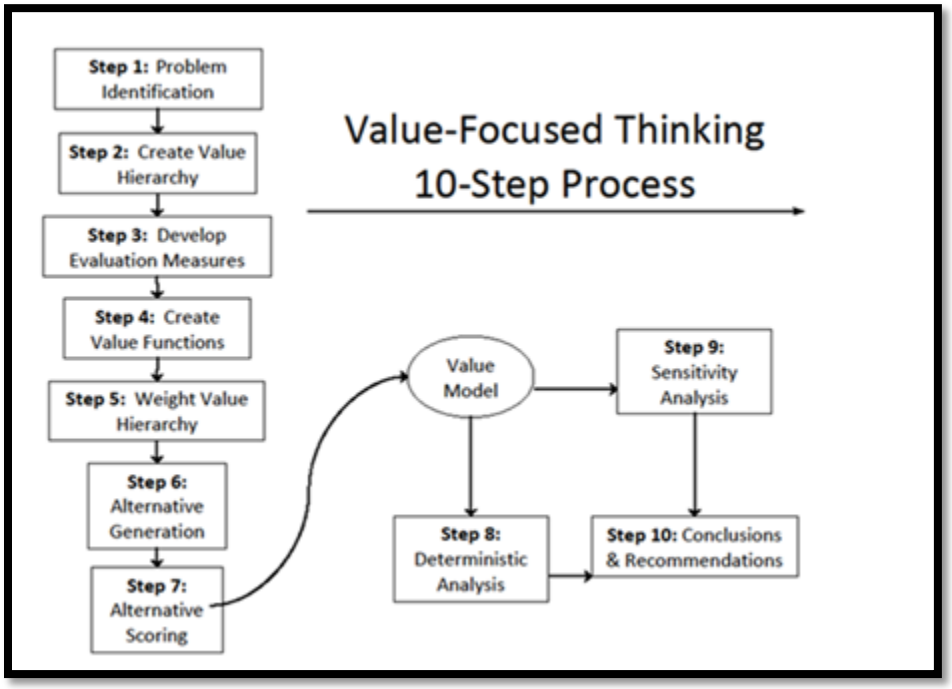


Figure 30. - VFT 10-Step Process (Shoviak, 2001)

An overview of the VFT process is as follows:

Step 1: Problem Identification

The first step in the VFT process is to fully understand the breadth and scope of the problem, in order to shrink the decision space to the context of interest. This problem definition helps make the decision process transparent, and it allows all of the participants a clear

understanding of what is expected. This includes defining a clear and impactful vision statement, an idea about the perspective of the problem, and the overall scope of the issue and what needs to be solved.

Step 2: Creating the Value Hierarchy

The value hierarchy is one of the main innovations and the structural backbone of the VFT process. It is the goal of the analyst to create a hierarchy that spans the decision space but also divides it independently. Therefore, it must be collectively exhaustive and mutually exclusive. These two properties ensure that the hierarchy is weighted correctly and precisely. The two main methods of developing this hierarchy are the top-down approach, which decomposes the problem into its values and objectives, and ultimately its measurements, and the bottom-up approach, which uses an exhaustive list of responses and the analyst attempts to combine them logically in groups that explain their merit. This is the approach that is done in this research. Value Hierarchies in this research were developed using Dr. Jeffery Weir's Value Hierachy Excel Spreadsheet macro.

Step 3: Developing Evaluation Measures

The measures of the hierarchy represent the lowest row of the hierarchy that feed into the values. These measures are the ultimate objective measurements that must be empirically collected from the system or process that is under review. In the lexicon of VFT, each individual row of the hierarchy is known as a tier, and each of the groups of values and measures is known as a branch. These measures can be constructed from latent variables or can be directly interpretable from real-world phenomena. The dividing terms of measures are directly measurable, measured by proxy, on either a natural or constructed scale. The main goal is to

accurately describe the objective attainment of the value the measure is feeding into. The following is a list of the types of measures used in VFT.

Table 6. - Types of Measures used in VFT

Types of Measures	Definition
Natural	Scale readily interpretable and in use
Constructed	Developed for a specific problem to measure attainment of objective
Direct	Directly measures degree of reaching the objective
Proxy	Does not directly measure degree of attainment, only shows correlation

Step 4: Creating Value Functions

After the creation and determination of measures for attainment of each specific value, value functions must be created for assessing the relative importance of the levels within the measures. Depending on the measurement system for the measures and the type of data that is collected for that measure, the value functions can either be categorical or continuous. These value functions are functions that transform the empirical data into utility measurements that are of importance to the Decision Maker. Therefore, the DM must be interviewed to assess their preferences and their risk biases and these functions must be monotonically increasing from 0 to 1 (Kirkwood, 1997). If this monotonic requirement were violated, utility preferences would be ambiguous, and it would be difficult to optimize the value measure to the best possible measure.

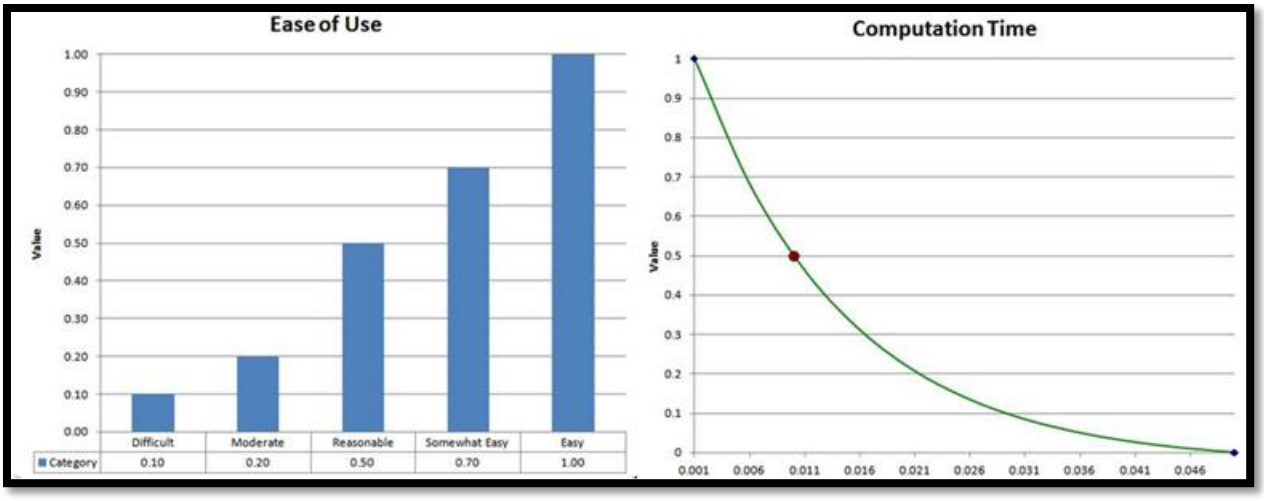


Figure 31. - Categorical and Continuous Value Functions

Step 5: Weighting the Value Hierarchy

After the development of the value functions, the individual values and measures must be weighed to enumerate their importance within the hierarchy. These should reflect the level that the measures help determine the actual value and each value weight should be reflective of how likely these values would lead to the overall determination of the alternative preference. Two different frameworks for calculating the weights can be used, local weighting and global weighting. The local weight of a value is the measure of importance for reaching the value in the tier directly above it. Global weighting is the overall importance of that value within the hierarchy, and these values can be analyzed to see if the ranking of values are slightly wrong or should be updated. Within the construct of local weighting, the weights that feed into the same node need to add up to one. It is an easy conversion to go from local to global weighting, as all of the local weightings along a branch should be multiplied together to calculate the global weight.

Step 6: Generating Alternatives

The generation of alternatives is an important step to the process and must be carried out repeatedly parallel to the main value hierarchy process. There are many ways to generate

alternatives from the values within the hierarchy and VFT allows a group the opportunity to both synthesize new alternatives and throw some out due to infeasibility or domination by other alternatives. If one alternative does not perform as well as another in all categories, it is considered to be dominated by that alternative. More information about generating alternatives has been covered in Chapter II.

Step 7: Scoring Alternatives

After the generation of feasible alternatives comes the calculation of the overall VFT value using the measures and value functions used within the hierarchy. For this research, after all response variables have been collected, they will be input into the hierarchy for each specific alternative. The output of the VFT value will be used to rank alternatives and assess their feasibility. A large amount of deliberation could be necessary in order to decide where each of the alternatives falls along the measure distributions.

Step 8: Deterministic Analysis

Deterministic Analysis stems from the scoring of the alternatives in Step 7. This is the combination of all of the alternative scores into one measure by taking the score of the individual evaluation measure one at a time and multiplying it by the global weight associated with that score. These are then added together to get the overall value for an alternative. It is a linear combination of global weight of the measure score and the actual score at the measure. This final measure can be compared against other factors outside of the hierarchy to see different relationships between alternatives and possibly generate other, new ideas for alternatives.

Step 9: Sensitivity Analysis

Sensitivity analysis is used to change the fundamental assumptions that are used in the model, such as the value functions and weightings, and see how the overall decision would

change for these tweaks of the model. This could show that the alternative choice is not generalized for all possible realities but only to the specific construct of the model. This is a type of meta-analysis that can both show the specificity of the choice in alternatives, and thus the robustness of that choice to random events that could change the weightings in the future, and it could help analyze the strength of the hierarchy, as stronger hierarchies should be more robust to slight changes. This is akin to the variance that occurs when a model fits to different training datasets. This step is useful in the Air Force, as often times, new leaders step into positions over analysts, and preferences for an alternative should be robust enough to fit nicely with those new DMs.

Step 10: Conclusions and Recommendations

This step is a communications and presentation based step that relies heavily on the ability to interpret and argue for the results of the analysis. These results are presented by the analyst to the Decision Maker, but should not be considered as the end-all answer to the problem. The Decision Maker should interface with the analyst in order to weight the alternatives against additional exogenous factors, such as cost and often time to complete these alternatives. At this step, other alternatives may be formulated using comparisons with similar already assessed alternatives (McGee, 2003).

VFT Hierarchy

Figure 32 is a representation of the Value-Focused Thinking Hierarchy that is used in this analysis. This hierarchy is utilized in order to provide a decision maker the means for making a valid, organized, though-out decision for choosing the correct classification algorithm that they need in each particular instance. The hierarchy is broken down into levels of values which constitute particular features that are of interest for the decision maker. These values can be

broken down further into additional levels for a higher resolution of clarification or broken down into measures that are objective measurements for certain variables that are estimated for the classification algorithms. These measures constitute the most important and highest leverage features of the algorithms that carry the most weight within the decision process. They also provide an easily accessed and interpretable measurement system that can be assessed to determine which features within the performance of the algorithm need to be improved or leveled with other features. For each individual measure, there is a weighting that is used per the subject matter experts opinion in order to provide subjective knowledge that helps judge the classification algorithm appropriately. Additionally to the weighting, there are value functions that determine the correct values that are assigned to each individual measurement's performance. The following VFT Hierarchy will be used in this research. Each branch will be explained in further detail later in this chapter.

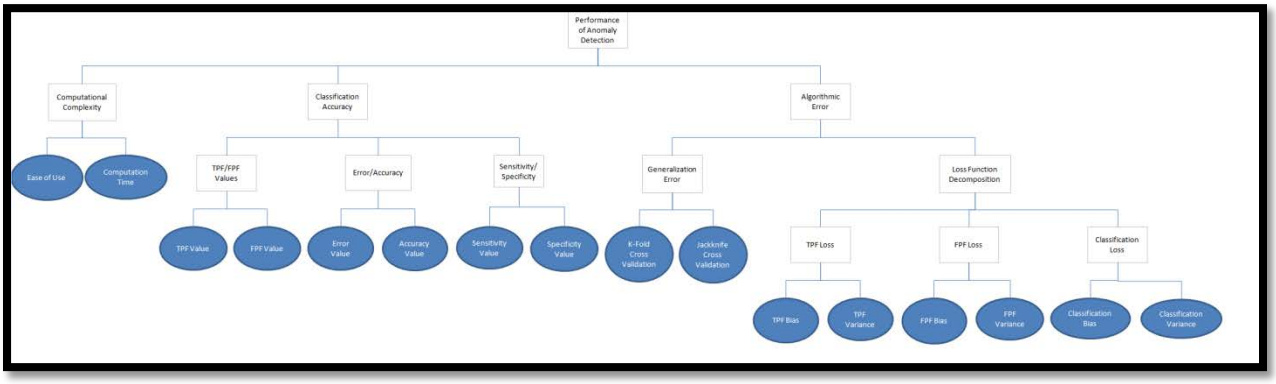


Figure 32. - VFT Hierarchy

First Experimental Design

A set of two Multivariate normal sample populations were developed using random draws from the Multivariate normal distribution using Matlab. Within each individual trial, the two sample populations represented a set of Background pixels and Target pixels for a hyperspectral imaging anomaly detection problem. For each pixel, a reflectance vector was

simulated with the x-axis representing the reflectance within the first discretized wavelength, and the y-axis representing the reflectance of the pixel within the second wavelength. These sample background and target pixel populations were created by varying different factors and then placed in a table in order to estimate and optimize the effects of each individual factor on different response variables. Since this toy study is simulating the anomaly detection methodology for a true HSI problem, the factors are representative for such a problem.

Experimental Factors

The first factor is the actual classifier that is used. These include the Quadratic Discriminant Analysis classifier, the Classification Tree, and the Naïve Bayes classifier. The second factor is the Mahalanobis Distance factor, which represents the distance from the centroids of the target distribution to the background distribution. This factor was divided into Short and Long levels, with Short pertaining to distances less than 5 and Long pertaining to distances greater than 5. The second and third factors pertain to the Covariance matrices of the target and background, respectively. The Target covariance matrices will result in distributions that are most representative of targets in HSI data, making the distributions dense with less variance than the background. The Background covariance matrix makes the distribution result in a higher variance and less dense than the target. The angle that each distribution will face each other is also varied using these matrices. The final factor is the Percentage of Target Pixels. This factor is representative of the HSI image data, as targets are typically sparse in the data. The levels of this factor are 1%, 5%, and 10% of the total number of pixels. These factors are seen in the following table.

Table 7. - Classification Algorithm Alternatives

Alternative Classification Algorithms
Quadratic Discriminant Analysis
Naïve Bayes Classification
Classification Trees

Table 8. - Factors used in Experiment

Mahalanobis Distance	Target Covariance Matrix	Background Covariance Matrix	Percentage of Target Pixels
Long >10	TCM1	BCM1	1%
Short <10	TCM2	BCM2	5%
	TCM3	BCM3	10%

Figure 33 represents a sample problem of the Target distribution, in blue, and the Background distribution, in red. As stated previously, the target distribution will have significantly less pixels than the background distribution. The large black points in the figure represent the centroids of the distributions, and the Mahalanobis distance will be calculated between these two centroids.

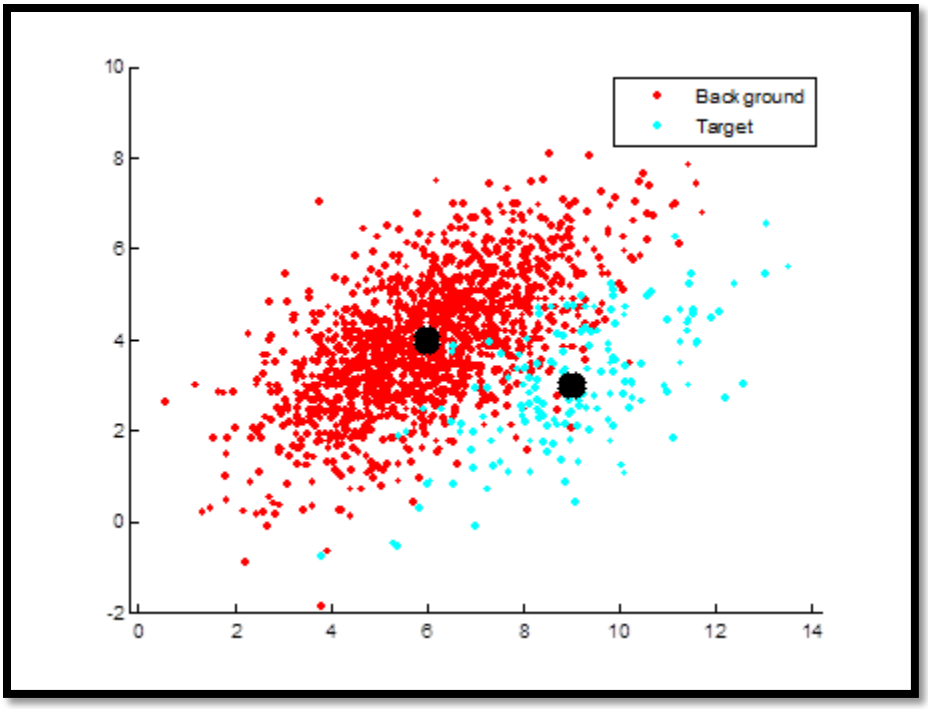


Figure 33. - Target and Background Distributions

Three different Supervised Classification Algorithms were used, including Quadratic Discriminant Analysis, Classification and Regression Trees, and Niave Bayes Classification. The responses that were collected included Computational Time and Effort values, True Positive Fraction (TPF), the False Positive Fraction (FPF), Sensitivity, Specificity, Accuracy, Precision (depending on context), a TPF and FPF Bias error value, a TPF and FPF Variance error value, a Domingos Classification Bias Error, a Domingos Classification Variance error, a k-fold Cross Validation Error value, and a Jackknife Cross-Validation Error value. Additionally, the value from the constructed Value Focused Thinking Hierarchy, after subjected to the value functions and weightings, was recorded for each combination. These values were compared for each specific classifier, and an overall value was computed by weighing the Hierarchy Values by the number of pixels in the image, to assess contextual information.

Classification Algorithms

The three classification algorithms used in this research are the Naïve Bayes Classifier, Classification Trees, and Quadratic Discriminant Analysis. These three algorithms were discussed in detail in Chapter 2, and these three plots are meant as examples. The first plot is an Example of the Naïve Bayes Classifier, which uses posterior estimates of class assignments based directly from prior distributions of probability estimates for each feature independently. It assumes the features are independent and calculates probability contours using the pdfs for each feature independently, which are computed around the centroid of the distribution. This can be seen in Figure 34.

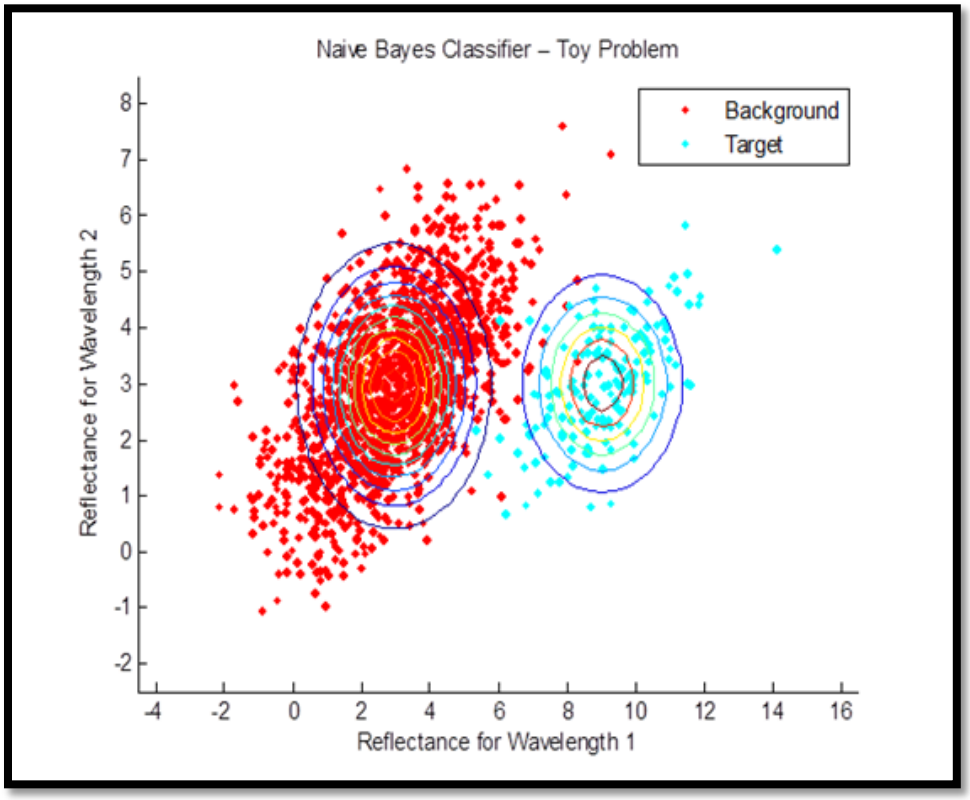


Figure 34. - Naive Bayes Classification

Figure 35 is a representation of the Classification Tree algorithm, which is comprised of nodes in which a decision is made based on the feature and the values of that feature that would

decrease the amount of information entropy the highest. Therefore, it divides the classification space into orthogonal class regions based on splits of the features.

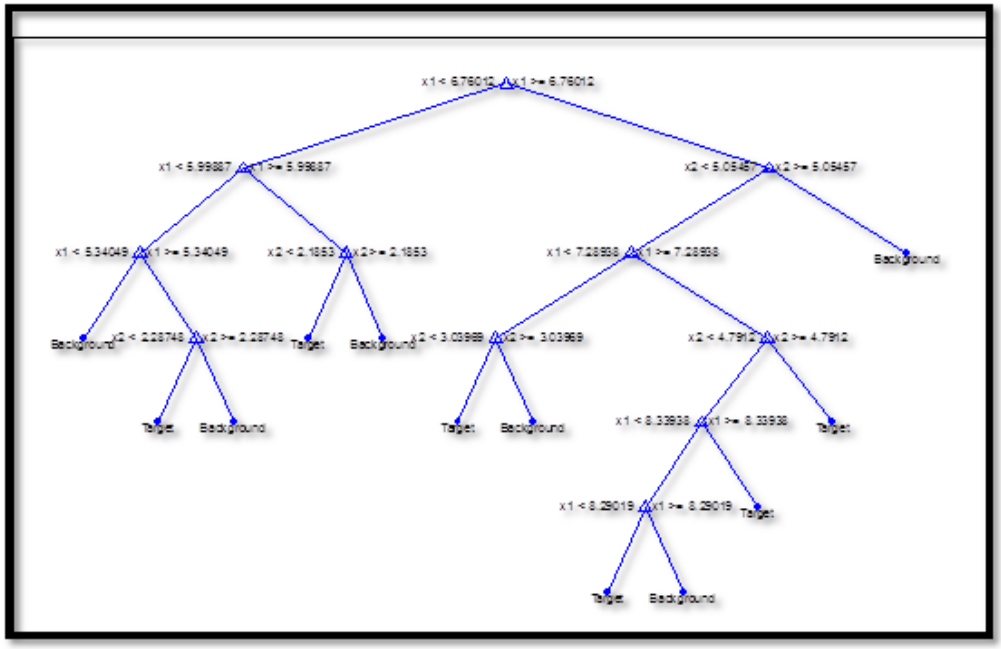


Figure 35. - Classification Tree Example

The final algorithm is the Quadratic Discriminant Classifier which utilizes the Mahalanobis distance between the two class distributions and calculates the likelihood that the point is either in one class or the other based on this distance. Therefore, the true covariance matrix of the class distributions is an integral part of the equation for the likelihood of class ownership. For various types of covariance matrices, which ultimately determine the shape of the distribution, the discriminant boundary will be different conic sections. This type of analysis has lower bias and higher variance than linear discriminant analysis, as the training sets that are used will determine the shape and location of the discriminant functions.

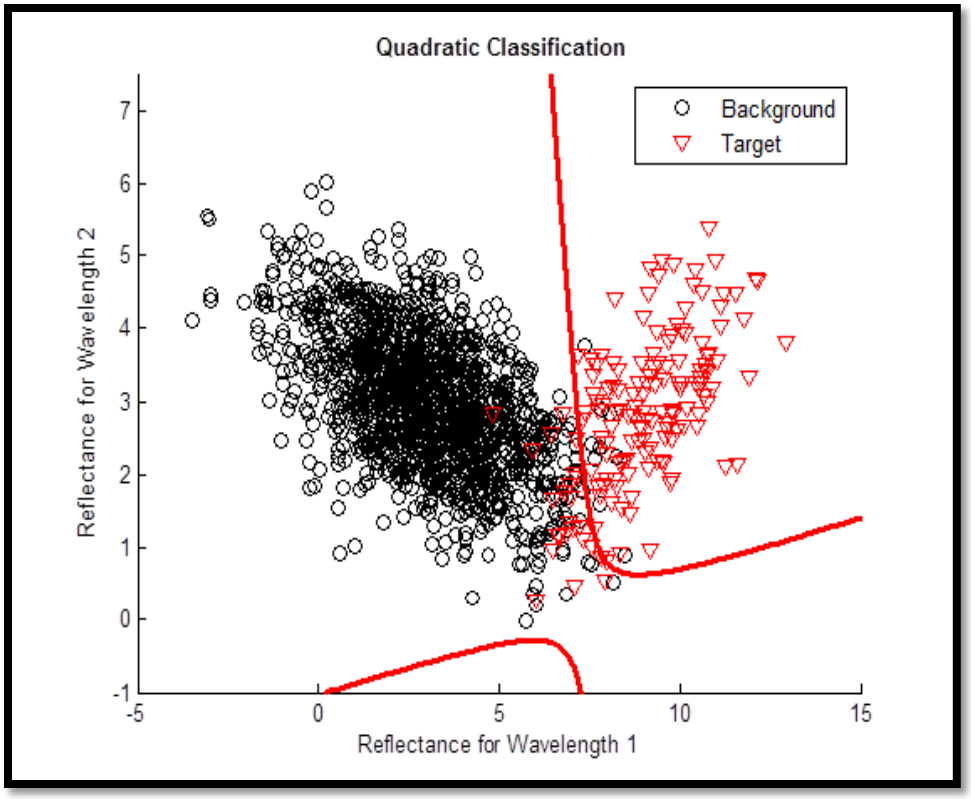


Figure 36. - Quadratic Discriminant Analysis

Mahalanobis Distance

The following is a representation of the difference between Euclidean distance and Mahalanobis Distance. Euclidean distance around a mean in a multivariate distribution will be measured from by the radius of the circle of equal distance that surrounds the centroid to the point of interest. This is seen below. The main issue with this type of distance is that it does not incorporate the covariance between the dimensions in the distribution, and thus, acts as if the dimensions were independent. Therefore, to assess outliers, it cannot be determined that a point in one dimensional direction could have a different actual distance from the mean than the other direction, when accounting for the covariance. A point in the dimension with greater variability will be interpreted as being far from the mean, even though it may still be in the distribution. This is the naivety of the assumption of independence. The points A and B here are the same Euclidean distance from the mean.

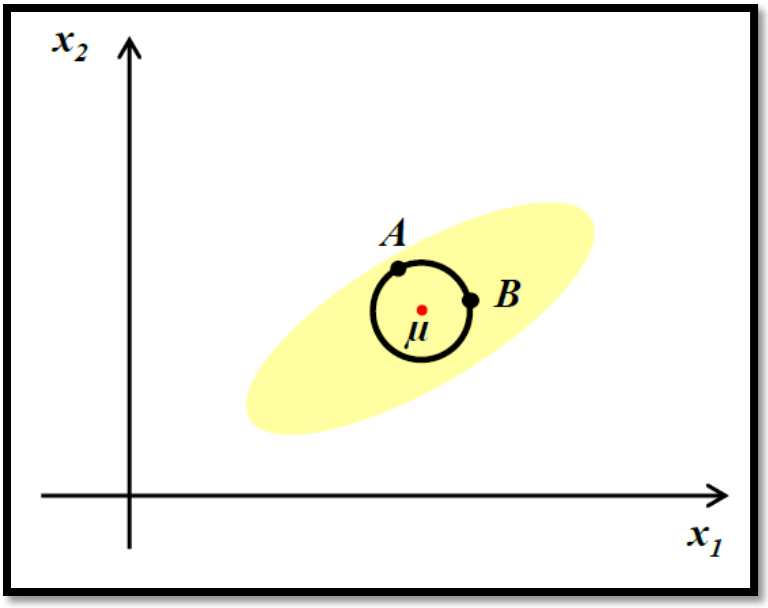


Figure 37. - Euclidean Distance (Tomaselli et al., 2013)

Mahalanobis distance accounts for this issue of distance from the mean in various directions by incorporating the covariance of the distribution within its calculation. Therefore, it does not assume that each dimension is independent, and for a two-dimensional case, equidistant points are now represented by ellipses with the axes corresponding to the variance in that direction. This allows the detection of outliers in one direction that may have been interpreted as the same distance when using the Euclidean definition. The points A and B here are the same Mahalanobis distance from the mean. This type of distance dilation is used primarily in the RX algorithm and its variants. The equation for Mahalanobis distance is the following (Tomaselli et al., 2013):

$$D_m(x,\mu) = \sqrt{(x-\mu)\Sigma^{-1}(x-\mu)^T} \tag{48}$$

where

Σ^{-1} is the inverse Covariance Matrix of the data

μ is the mean of the distribution

x is the point of interest.

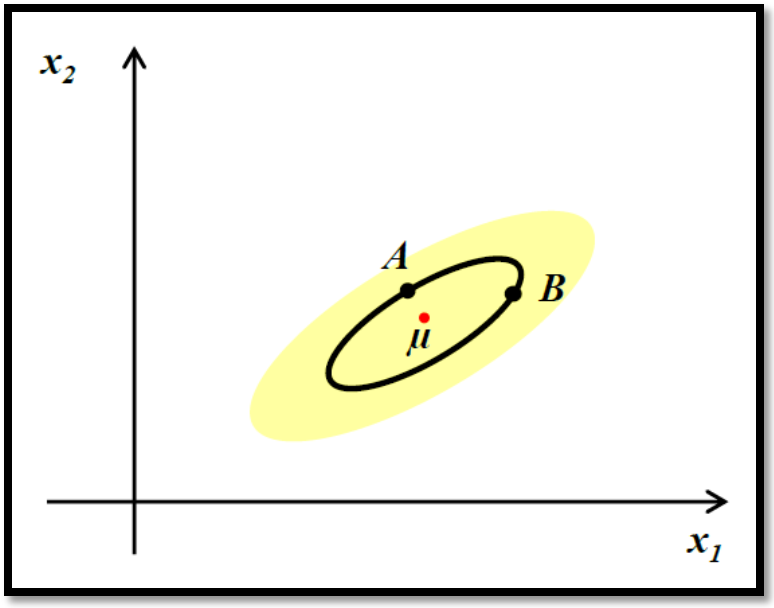


Figure 38. - Mahalanobis Distance (Tomaselli et al., 2013)

Experimental Measures/Responses for the VFT Hierarchy

Transitioning back to the VFT hierarchy methodology, Table 9 represents the values and their associated responses for Computational Complexity and Classification Accuracy. Most of the measurements are natural and direct, although the Ease of Use measurement uses a constructed scale to define the level of triviality in the algorithm.

Table 9. - Measures for Computational Complexity and Classification Accuracy

	Responses/Measures of Interest							
Value	Computational Complexity		Classification Accuracy					
Sub-Value			TPF/FPF		Error/Accuracy		Sensitivity/Specificity	
Responses/Measures	Ease of Use	Computational Time	TPF	FPF	Error	Accuracy	Sensitivity	Specificity
Type	Constructed, Direct	Natural, Direct	Natural, Direct	Natual, Direct	Natural, Direct	Natural, Direct	Natural, Direct	Natural, Direct

The branch for Computational Complexity is divided into Ease of Use and Computation Time measurements.

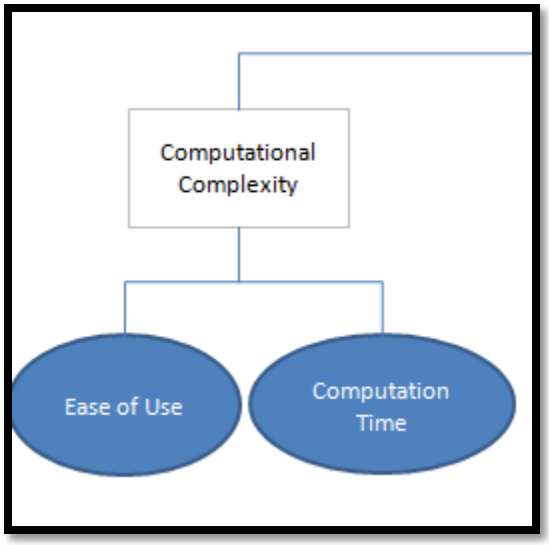


Figure 39. - Computational Complexity Branch

The branch for the Classification Accuracy value is split into three “means objective values”, which separate the use of the confusion matrix into three constituent parts (TPF/FPF Values, Error/Accuracy Values, and Sensitivity/Specificity Values) depending on the application that the analyst would like to work and deliver results to the Decision Maker in. This helps facilitate conversation and decreases the noise when briefing superiors. Each of these means objective values can be turned on or off and usually only one would be used in a typical application, as there is no real need to compare each of the measurements. This branch can be seen in Figure 40.

The following graph depicts the confusion matrix for a two-class classification problem, with the predicted number of positive, negative, and total classifications across the column space, and the actual number of positive, negative, and total data points placed across the rows.

Table 10. - Confusion Matrix

		Predicted		
		Positive	Negative	Total
Actual	Positive	TP	FN	p
	Negative	FP	TN	n
	Total	p'	n'	N

Additional measures that can be assessed within a two-class classification problem are found in the Table 11. They include an estimate of total error, accuracy, precision, and Sensitivity, which is equal to the True Positive Fraction for a two-class classification problem, and Specificity, which is the difference between unity and the False Positive Fraction.

Table 11. - Confusion Matrix Formulae

Name	Formula
(total) error	(FP+FN)/N
Accuracy	(TP+TN)/N
FPF, false positive fraction	FP/n
TPF, true positive fraction	TP/p
Precision	TP/p'
Recall	TP/p (=TP fraction)
Sensitivity	TP/p (=TPF)
Specificity	TN/n (=TNF=1-FPF)

Table 12 is a representation of the typical Matlab output for the Confusion matrix. True Positives (TP) are in the upper left element, while False Negatives (FN) are in the upper right element. In the bottom left element are False Positives (FP) and in the bottom right are True Negatives (TN).

Table 12. - Matlab Confusion Matrix Output

		Predicted	
		Predicted Positive	Predicted Negative
Actual	Actual Positive	1435	5
	Actual Negative	14	146

The representation of the Classification Accuracy branch is seen below.

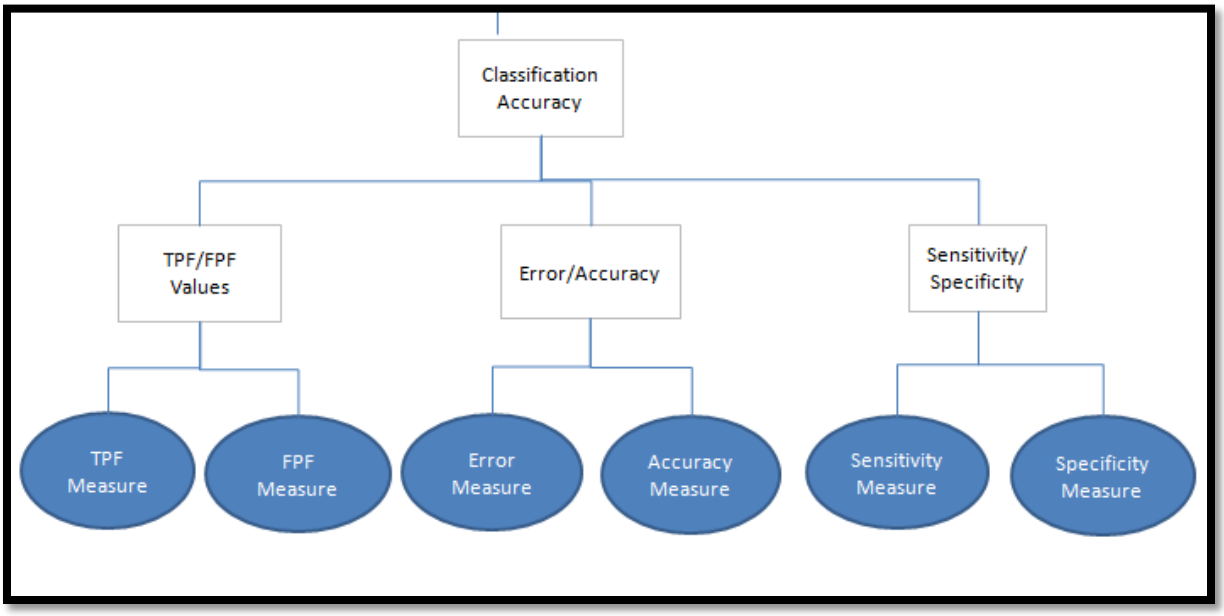


Figure 40. - Classification Accuracy Branch

The Algorithmic Error value branch is split into Generalization Error and Loss Function Decomposition Means Objective Values. The Generalization Error is computed using both K-Fold Cross Validation Error, in which the K will be determined by quick preliminary testing, and Jackknife Cross Validation Error, in which each point is held out as the test set and the rest of the data is treated as training data to train the algorithm and predict that point. The amount of times that the point is misclassified is integrated as the error. For the Loss Function Decomposition, two types of decompositions were conducted, one based on posterior estimates of the TPF and FPF values, using an MSE Quadratic loss function that is typical in regression, and thus it treats the TPF and FPF value functions, along with the classifier, as part of the same function, which can be represented on a continuous scale. For the classification loss of classifying each individual point as target or background, Domingos' unified decomposition for general loss functions will be applied. This comparison of loss function performance is a novel approach

taken in this research. Although there is some debate, I have determined that the Generalization Errors and Variances are all Natural, Direct measures, while the Biases are all Natural, Proxy measures, as the estimate of bias is in fact being used to determine the accuracy of the classification system using a difference between the expected parameter value and the optimum parameter value. This calculation of expected parameter value brings the measurement out of the direct framework to the proxy framework. Variance, however, is directly determined by the natural variation between the expected output and each individual output. These definitions are largely notional in this case and other interpretations may suffice.

Table 13. - Measures for Algorithmic Error Response

	Responses/Measures of Interest							
Value	Algorithmic Error							
Means Obj-Value	Generalization Error		Loss Function Decomposition					
Sub-Value			TPF Loss		FPF Loss		Classification Loss	
Responses/Measures	K-Fold Cross Validation Error	Jackknife Cross Validation Error	TPF Bias	TPF Variance	FPF Bias	FPF Variance	Classification Bias	Classification Variance
Type	Natural, Direct	Natural, Direct	Natural, Proxy	Natural, Direct	Natural, Proxy	Natural, Direct	Natural, Proxy	Natural, Direct

The following figure is a representation of the Algorithmic Error Branch.

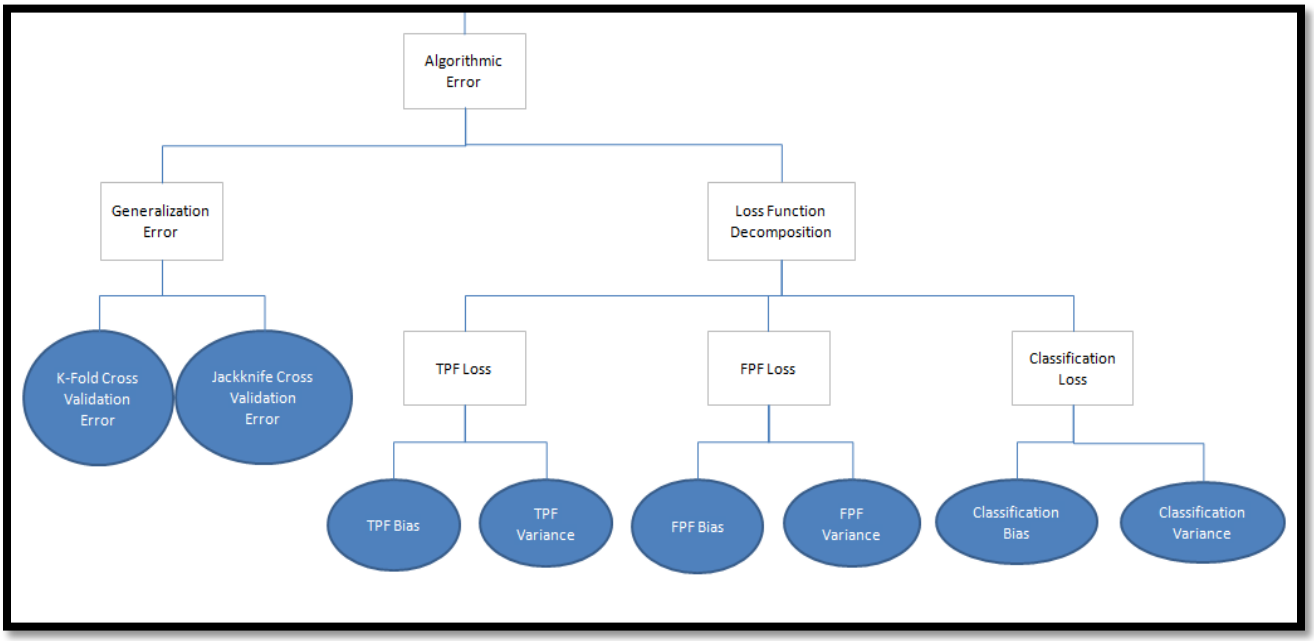


Figure 41. - Algorithmic Error Branch

Bootstrapping

A hybrid Parametric/ Non-Parametric bootstrapping approach will be utilized to generate results for the Loss Function Decomposition Means Objective Value. Parametric bootstrapping is the process of simulating a new set of feature data from the empirical distribution of the feature data and then using this simulated data to draw class or response data from the conditional distribution of $(\hat{y}|x)$ where $\hat{y}$ are the predicted classes for each data point in x. This is demonstrated in Figure 42, as data is first used to fit a model, and then the fitted model is sampled, with replacement, with new simulated data to generate a new estimate.

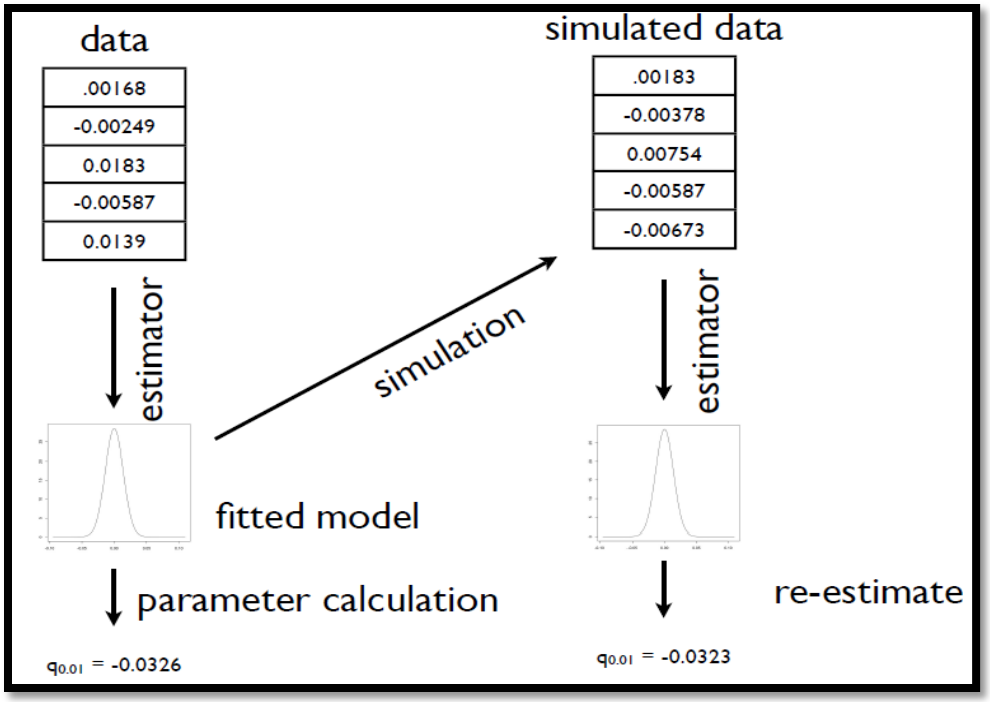


Figure 42. - Parametric Bootstrapping (Shalizi, 2011)

Non-parametric bootstrapping includes the upfront resampling, with replacement, of both the response values and the feature values. This in essence is treating the original set of data as if it were a complete population, and each new sampled data set is just a sample from the overall population, after which a parametric model can be applied to calculate a re-estimate of the parameter of interest. The difference between these two formulations is just a matter of the sequence of parameter estimation, with parametric bootstrapping having that estimation come before the simulation of the data, and non-parametric having it come after (Shalizi, 2011).

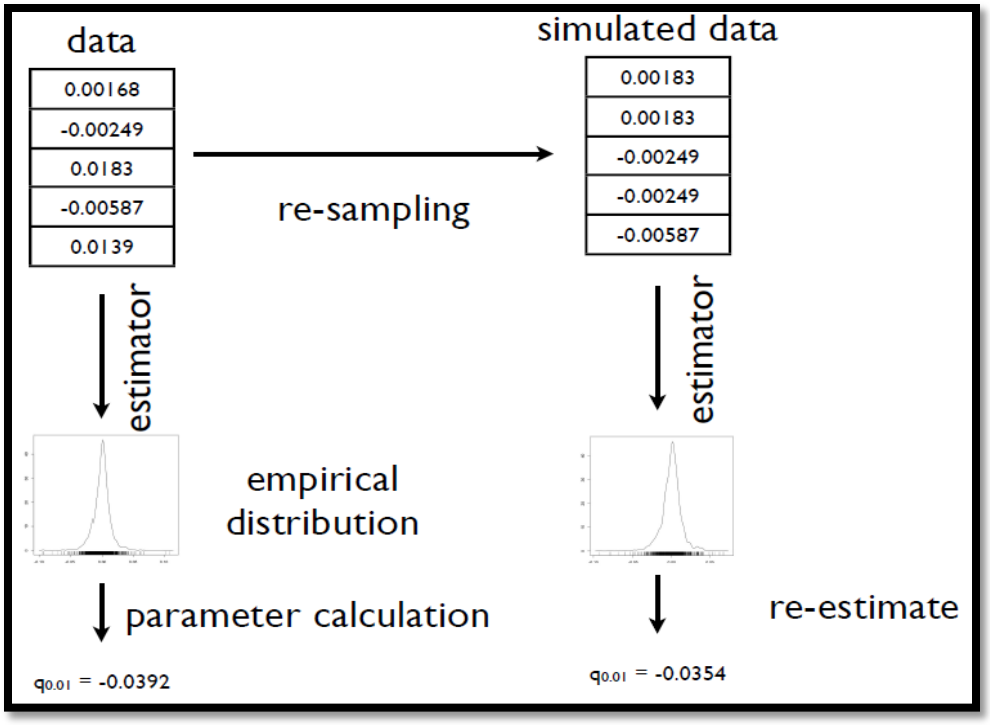


Figure 43. - Non-Parametric Bootstrapping (Shalizi, 2011)

There is an additional methodology for bootstrapping known as residual re-sampling that holds fixed a deterministic function of feature input to class or response output and then adds the residual value that accounts for the stochastic noise by resampling it, with replacement, from the original conditional probability distribution. In this research, both the x and y values will be resampled, which is also known as bootstrapping the indices. This allows the maximum separation of the performance of the classification algorithm and associated confusion matrix parameter of interest from the performance of the resampling methodology.

Cross-Validation

Cross-Validation will be utilized for the measurement of generalization error to different sets of data. This process splits the original training set of data into different partitions in various ways. K-fold Cross-Validation is a way of splitting the training set into "folds" by partitioning it into K

different data sets. Each of these data sets will take turns as the test dataset in the classification. The rest of the folds will be used as training sets to train the algorithm. Once trained, error will be calculated by using this trained algorithm against the fold that is representing the test dataset. This is done for each fold and then each of these errors of misclassification percentages is averaged to calculate the measurement of error.

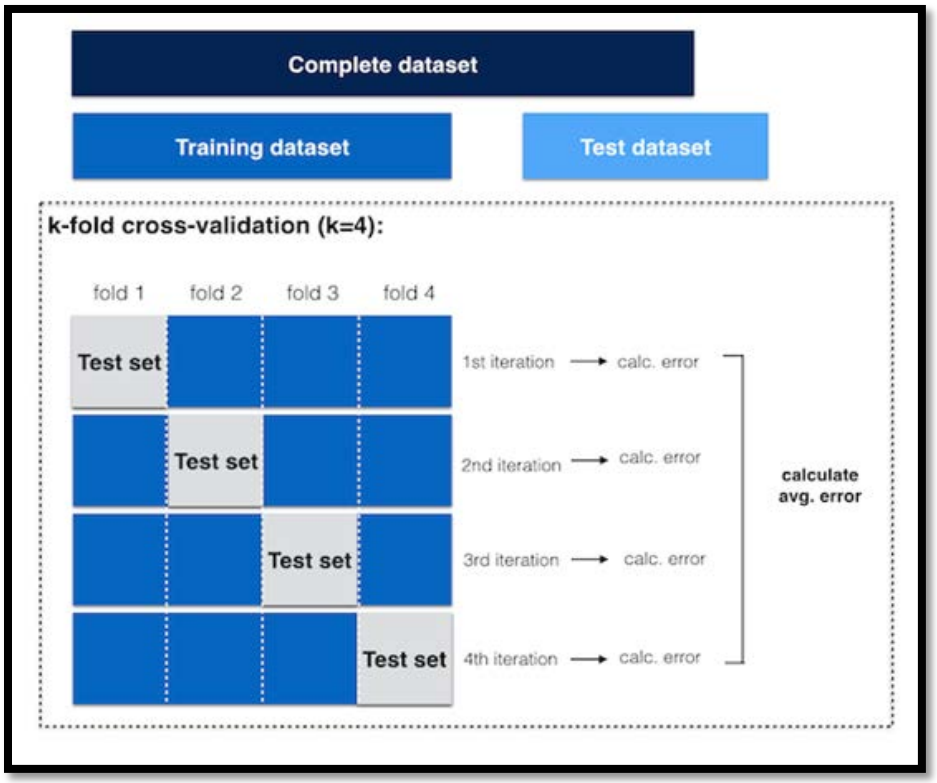


Figure 44. - k-fold Cross-Validation (Raschka, 2015)

Jackknife Cross-Validation will also be used. This is similar to K-Fold Cross Validation, but now the number of data points, n, is used as the number of folds, k. This means that each data point will be used as a test dataset while all of the other data is used to train the algorithm. The percentage of points misclassified will be used as the measure of error.

HSI Data

The data that has been vetted in these experiments is from a program known as the Hyperspectral MASINT Support to Military Operations program (HYMSMO) using the Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor. This experiment is a simulation of Airborne Reflective Emissive Spectrometer (ARES) and specifically are part of the Forest Radiance I and Desert Radiance II experiments. The HSI data is taken at 210 different wavebands in the visible and IR ranges of the EM spectrum and the number of total pixels, target pixels, and neighborhood pixels have been counted and recorded. Various numbers of targets are present in each of the scenes and the target percentage of the percentage of number of target and neighborhood pixels to total pixels is recorded. Some of these targets are synthetically placed into the scenes. The actual images are displayed below. Images ARES1, ARES2, ARES3, ARES2D, ARES2F, ARES3D_10k, and ARES3F are used in the first HSI experiment. ARES1D, ARES1F, and ARES4F are used in the Validation experiment.

Table 14. - ARES Image Factors

Image	Bands	Total Pixels	Target Pixels	Neighborhood Pixels	Total Targets	Target Percentage	Test
ARES1	210	26196	237	0	6	0.00904718	Intital HSI
ARES2	210	18810	321	0	8	0.01706539	Intital HSI
ARES3	210	16588	152	0	8	0.00916325	Intital HSI
ARES1D	210	57909	235	437	6	0.01160441	Validation
ARES1F	210	30560	1007	973	10	0.06479058	Validation
ARES2D	210	22360	523	1942	46	0.1102415	Intital HSI
ARES2F	210	47424	307	1221	30	0.03221997	Intital HSI
ARES3D_10k	210	11024	157	112	4	0.02440131	Intital HSI
ARES3F	210	30736	145	314	20	0.01493363	Intital HSI
ARES4F	210	16400	109	339	29	0.02731707	Validation

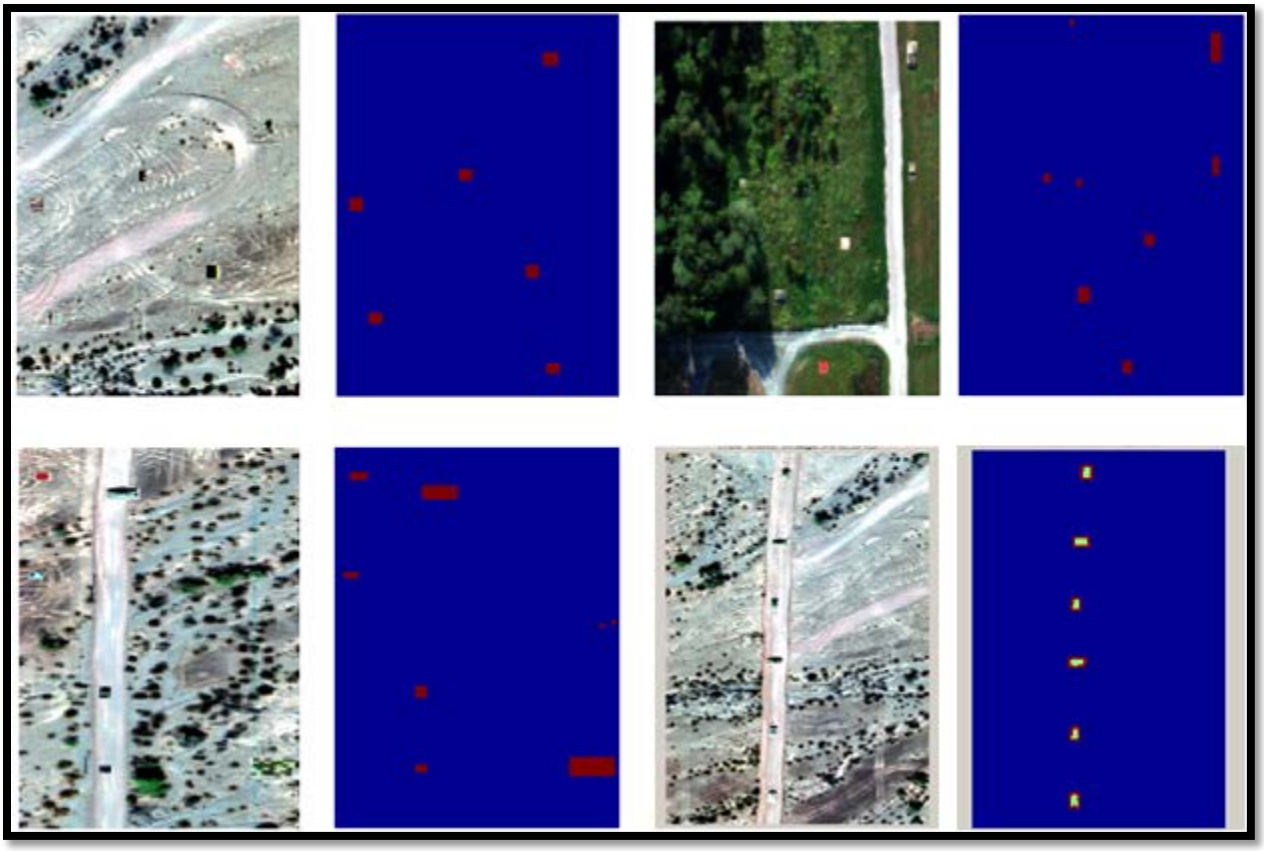


Figure 45. - ARES Images 1, 2, 3, 1D (Orloff et al., 2000)

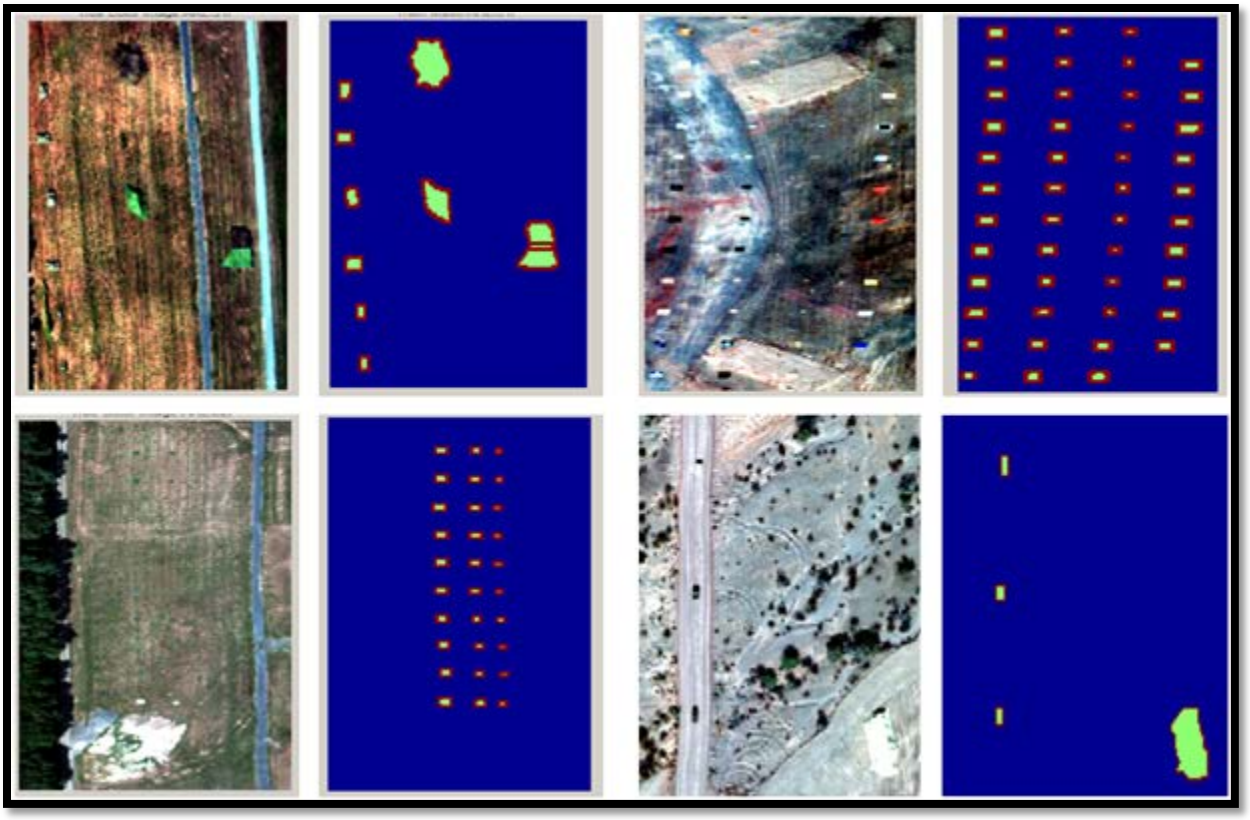


Figure 46. - ARES 1F, 2D, 2F, 3D10K (Orloff et al., 2000)

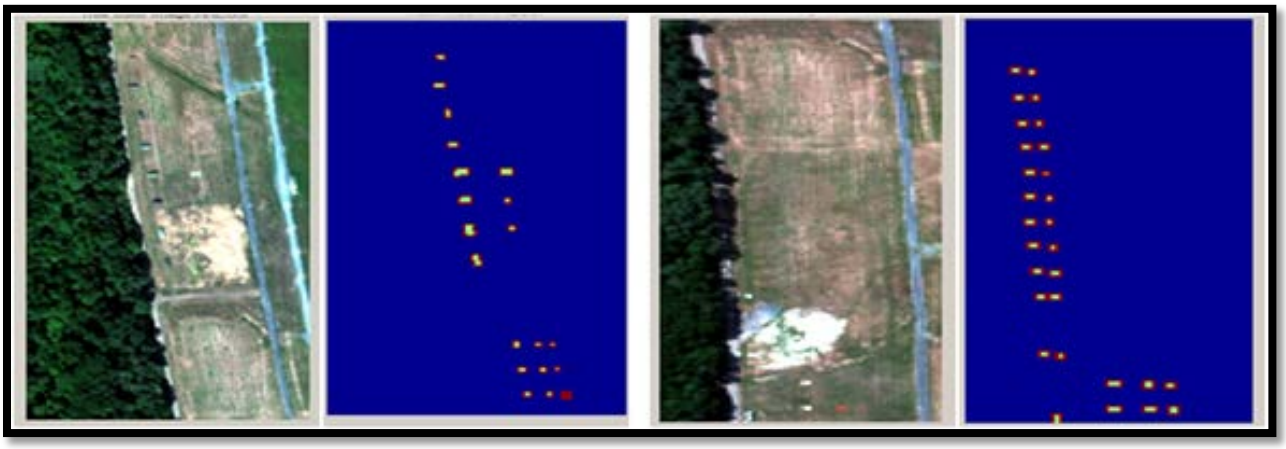


Figure 47. - ARES 3F, 4F (Orloff et al., 2000)

Factors for HSI Data Experiment

In order to make the VFT hierarchy values more representative to real world scenarios, an experiment will be carried out using the same three algorithms as before but now on ARES images. Principal Component Analysis will be used for Feature Extraction to reduce the number

of features from the 210 bands in the images. The Supervised Classification Algorithms will then be applied to the image data and the same responses will be collected as before, which will be entered into the VFT hierarchy. A post-processing weighting will be used to determine the final value that will determine which algorithm performed most satisfactorily, and thus, in VFT parlance, is the best alternative.

Table 15. - HSI Data Experiment

Algorithm	Images
Quadratic Discriminant Analysis	ARES1
Classification Trees	ARES2
Naïve Bayes	ARES3
	ARES2D
	ARES2F
	ARES3D_10
	ARES3F

Application of Various Bias/Variance Frameworks for Classification

In this analysis, the goal is to compare and contrast various frameworks for the bias and variance decomposition of a loss function in terms of Mean Squared Error and in the Zero-One loss function for classification. The main sub-goal is to determine a optimum computation to enter into the Value Focused Thinking methodology for comparing supervised classifiers.

There have been many formulations for the bias and variance decomposition of loss functions in both regression and classification situations. Within the regression framework, the Mean-Squared Error Composition works best as it is somewhat simple to breakdown the MSE loss function in terms and bias and variance. In classification, it has been a bit more difficult and many formulations have been postulated. These include Friedman's formulation and Domingos' formulation (Friedman, 1997) (Domingos, 2000). An additional formulation for the

True Positive and False Positive Fractions using the traditional decomposition of the quadratic Mean Squared Error loss function has been developed herein. Matlab has been used on a few toy situations to compare these bias/variance calculations for classification.

Conceptual Definitions

The idea of bias stems from the intuitive idea of accuracy of a classifier. Within a certain data set, there is a target that must be reached in terms of fitting some parameter and having it fall near a pre-specified value. This is akin to the situation of playing darts. The player has an optimum target that they are reaching for, which in some cases is the bull's-eye. The distance that the player's throw of the dart lands away from the bull's-eye can be thought of as the bias in the system of the thrower's mechanics. In this sense, the bias is the error due to the difference between the expected prediction and the correct value that is being approached. For one instance of a training set and the resulting model analysis of the training set, only one distance is calculated from the target. When multiple training sets, as in bootstrapping, are used to train the model, the average distance is used away from the target. This target can either be comprised of an *a priori* probability of the prediction of certain combinations of features in the feature matrix or it can be some target that some estimation is attempting to approach. Such targets would include 1 in terms of True Positive Fraction or 0 in terms of False Positive Fraction. The randomness that is inherent in the data sets is what creates a range of predictions, making the process stochastic in nature and not deterministic. If the same training set was used for each replication, and the modeling algorithm had no random components, then the process would be deterministic.

The idea of variance can be thought of as the precision of the distribution of model estimates. In terms of the dart throwing example, the precision is how closely each throw of the

dart lands from one another when considering some target that the thrower is attempting to hit. In a classification setting, the algorithm will be trained on multiple training sets. When each unique combination of feature variable values is used to estimate the class of the data point, the variance arises in terms of how closely packed the estimated classes are. Since in a classification setting, the zero-one loss function is used to determine if the correct class has been calculated, the situation is slightly different from regression that has a continuous response variable, and thus an idea of distance between estimations. However, in terms of the posterior probabilities that a certain class has, the variance can be thought of as the precision between the different probabilities. If one class is being predicted a lot more than another class, even though the prior probabilities for those classes are close together, there would be large bias in that one case. If there are large fluctuations for the posterior probabilities of the class estimations, then there would be large variance for those points. For TPF and FPF, the idea is a bit more natural, as there will be an idea of precision based on the variance of the fractions that are generated. If these fractions change from training set to training set, the variance will be higher. Duda et al., states,

> Given that there is no general best classifier unless the probability over the class of problems is restricted, practitioners must be prepared to explore a number of methods or models when solving any given classification problem. The bias measures the accuracy or quality of the match: high bias implies a poor match. The variance measures the precision or specificity of the match: a high variance implies a weak match (Duda et al., 2001).

In terms of model fitting, there is a trade off of bias and variance due to the number of parameters that are used in the model and the flexibility the model has for predicting the classes of the unique feature space combinations. Domingos observed that both flexible learners with complex representations of parameters and basic learners are both seen to perform well in certain experiments, and sometimes these simple learners outperform the more complex ones. He states, “In recent years the reason for this has become clear: predictive error has two components, and

while more powerful learners reduce one (bias) they increase the other (variance). The optimal point in this trade-off varies from application to application" (Domingos, 2000). Analysis must be done in order to determine the optimum settings to trade off the probability of fitting each point exactly, which would be the bias portion of the error, and the ability for the model to generalize to other additional data sets, so it does not over-fit to one particular, unique data representation. When bias is decreased, the model is more flexible, and can fit each future data point well, however the variance will be increased, as that instance of the model will not be able to fit other future data sets well. On the other hand, when variance is decreased, the expected classification will have a larger discrepancy from the actual classification, but the model will perform better on the whole when exposed to many different data sets. Duda et al. reveals, "Designers can adjust the bias and variance of classifiers, but the important bias-variance relation shows that the two terms are not independent; in fact, for a given mean-square error, they obey a form of 'conservation law'" (Duda et al., 2001).

Bias and Variance for Regression (and TPF/FPF Values)

The breakdown for bias and variance for regression is seen in the following formulae (Kuncheva, 2004) (Duda et al., 2001):

$$\varepsilon_D[(g(x;D) - F(x))^2] = (\varepsilon_D[g(x;D) - F(x)])^2 + \varepsilon_D[(g(x;D) - \varepsilon_D[g(x;D)])^2] \quad (49)$$

$$F(x) = E[y|x] \quad (50)$$

$$bias^2 = (\varepsilon_D[g(x;D) - F(x)])^2 \quad (51)$$

$$variance = \varepsilon_D[(g(x;D) - \varepsilon_D[g(x;D)])^2] \quad (52)$$

where

D is an instance of a training data set

$g(x;D)$ is the model estimate of the specific x vector in the data set D

This representation will be used in this research for an approximation of TPF and FPF values by considering each D to be a bootstrapped sample from the original distribution of data. For TPF, the $F(x)$ value will be represented by the optimal value of 1, and for FPF, $F(x)$ will be represented by its optimal value, which is 0.

Domingos' Formulation

Pedro Domingos redefined the bias/variance decomposition for an arbitrary loss function, and showed that the decomposition specializes to the standard one for the squared-loss case, and to one that is similar to the Kong and Dietterich decomposition for the zero-one function (Kong et al.,1995). Domingos states that each of the previously published decompositions in the literature for the zero-one loss function is flawed. He states,

> None has a clear relationship to the original decomposition for squared loss. One source of difficulty has been that the decomposition for squared-loss is purely additive (i.e., loss=bias+variance), but it has proved difficult to obtain the same result for zero-one loss using definitions of bias and variance that have all the intuitively necessary properties. Here we take the position that instead of forcing the bias-variance to be purely additive, and defining bias and variance so as to make this happen, it is preferable to start with a single consistent definition of bias and variance for all loss functions, and then investigate how loss varies as a function of bias and variance in each case (Domingos, 2000).

Therefore, Domingos is attempting to flip the logic from trying to find bias and variance terms that match up exactly with the MSE function to definitions that could be set up to yield various types of loss functions. This is desirable since a bias-variance tradeoff issue exists in any form of generalization problem, and thus if Domingos succeeds, he can apply this idea to any logically sound loss function.

Domingos defines various definitions for different concepts that he uses to build his unified decomposition. He applies the loss function $L(t, y)$, where *t* is the true value for the certain *x* vector prediction, and *y* is the actual prediction using the classifier function. The first definition is for the concept of a main prediction.

Definition 1- Main Prediction:

The main prediction for a loss function *L* and set of training sets *D* is

$$y_m^{L,D} = argmin_{y'} E_D[L(y, y')]. \quad (53)$$

Domingos states,

> The expectation is taken with respect to the training sets in *D*, i.e., with respect to the predictions *y* produced by learning on the training sets in *D*. Let *Y* be the multiset of these predictions. (A specific prediction *y* will appear more than once in *Y* if it produced by more than one training set) (Domingos, 2000).

Domingos explains the idea of main prediction as thus, "The main prediction is the *y'* value whose average loss relative to all the predictions in *Y* is at a minimum. It is the prediction that "differs least" from all the predictions in *Y* according to *L*" (Domingos, 2000). In squared loss, this is the mean of the predictions, in absolute loss, it is the median, and under the zero-one loss it is the mode. It represents the "central tendency" of the learner. The next definition is for bias.

Definition 2- Bias:

The bias of a learner on an example *x* is $B(x) = L(y^*, y_m)$, where y^* is the optimal prediction, and y_m is the main prediction. This bias is the loss that is measured when the main prediction is compared to the optimal prediction. The next definition is the variance.

Definition 3- Variance:

The variance of a learner on an example *x* is $V(x) = E_D[L(y_m, y)]$. The variance is the average loss that is measured by the actual predictions relative to the main prediction. For all examples, the bias and variance can be averaged, and represented as $E_x[B(x)]$ and $E_x[V(x)]$. The following definition is for noise.

Definition 4- Noise:

The noise of an example x is $N(x) = E_t[L(t, y^*)]$ This is the part of the loss that is representative of the stochastic noise and is not dependent on the training set or learning algorithm. For most loss functions, these three values can be combined in the following formula.

$$E_{D,t}[L(t,y)] = c_1 N(x) + B(x) + c_2 V(x) \tag{54}$$

The c_1, c_2 values are multiplicative factors that will be different for various loss functions. For the quadratic loss function, this equation is valid as shown in the following theorem.

Theorem 1 – Squared Loss:

Equation 54 is valid for squared loss, with

$$c_1 = c_2 = 1 \tag{55}$$

$$y^* = E_t[t] \tag{56}$$

$$y_m = E_D[y] \tag{57}$$

$$E_{D,t}[(t-y)^2] = E_t[(t - E_t[t])^2] + (E_t[t] - E_D[y])^2 + E_D[(E_D[y] - y)^2] \tag{58}$$

$$Noise = E_t[(t - E_t[t])^2] \tag{59}$$

$$Bias^2 = (E_t[t] - E_D[y])^2 \tag{60}$$

$$Variance = E_D[(E_D[y] - y)^2] \tag{61}$$

Domingos' explains that this definition can now be used for classification, "We now show that the same decomposition applies to zero-one loss in two-class problems, with c_1 reflecting the fact that on noisy examples the non-optimal prediction is the correct one, and c_2 reflecting that variance increases error on biased examples but decreases it on biased ones" (Domingos, 2000). This issue reflects the multiplicative issue that has been seen for other decompositions of the zero-one loss function, in which the direction of the bias has an impact on how variance affects

the error. For this theorem, $P_D(y = y^*)$ is the probability over the training sets in *D* that the learner predicts the optimal class for *x*.

Theorem 2 – Zero-One Loss:

Equation 54 is valid for zero-one loss in two-class problems, with:

$$c_1 = 2 * P_D(y = y^*) - 1 \quad (62)$$

$$c_2 = 1 \text{ if } y_m = y^*, \quad (63)$$

$$c_2 = -1 \text{ otherwise} \quad (64)$$

About the uniqueness of the variance function, Domingos states,

> The fact that the variance is additive in unbiased examples but subtractive in biased ones has significant consequences. If a learner is biased on an example, increasing variance decreases loss. This behavior is markedly different from that of squared loss, but is obtained with the same definitions of bias and variance, purely as a result of the different properties of zero-one loss" (Domingos, 2000).

This means that there is a much higher tolerance for variance and it should be treated differently in these classification situations. This is because the increase of average loss that is a product of variance when dealing with unbiased samples is somewhat offset by the decrease when dealing with biased samples. Additionally, the c_1 value leads to another difference between quadratic loss and zero/one loss functions, in that in quadratic loss, the noise will always increase the overall error, but in zero-one loss, when the predicted class is not the optimal class, increasing the noise value actually decreases the overall error, and it is thus desirable to increase the noise to make better predictions in these situations (Domingos, 2000).

IV. Results and Analysis

Value Focused Thinking

The Value-Focused Thinking 10-Step Process was utilized as a methodology to select the best algorithm out of a set of three alternatives spanning the field of pattern classification. This type of analysis allows the analyst to make decisions that are founded on logic and can be repeated for future research or applied to different alternatives. It can easily be modified for different algorithms and different measures. It also is a very transparent process that can be easily communicated and understood by Decision Makers and other analysts. The process, once again, is shown below.

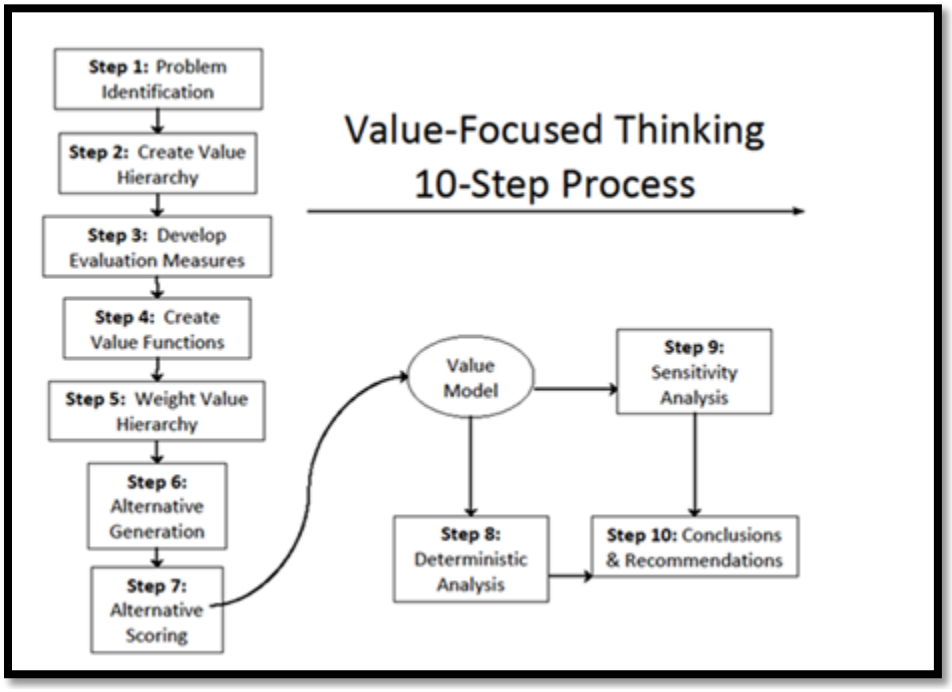


Figure 48. - VFT 10-Step Process (Shoviak, 2001)

Step 1: Problem Identification

The first step of the process is the scoping and understanding of the problem. In the previous chapters, the HSI target detection problem has been well flushed out and this understanding of the problem will be used for the formal analysis. Keeney states that developing vision statements, perspectives, and delineating the scope of the problem will allow the analyst to keep an eye on the values within the system. These three items are seen below.

- **Vision statement:** Be able to effectively **detect target from background pixels** using classification algorithms to provide the **most accurate and useful information** to a decision maker in a timely manner.

- **Perspective:** The perspective that will assess the accuracy and usefulness of the detector will be an interested decision maker that has a stake in target detection. This will vary per situation. SMEs will ultimately let me know if the **parameters and the certain type of detector is useful per the situation.**

- **Scope:** This research focuses on a few **types of detecting methodologies.** The problem will be analyzed the problem in terms of **variance and bias reduction for regression and classification**, usefulness for the **user**, and **classification accuracy** in different frameworks.

Step 2: Creating the Value Hierarchy

As observed previously, a hierarchy was developed that encapsulates both the values and measures of interest in this study. The hierarchy would then be weighed for each individual image per the three classifiers and combined using a weighting. This methodology is seen below. The weighting is reflective of the contextual information that is seen in the image. This is akin to ranking the images per the situation or the criticality in the scenario, which often happens during real target detection situations.

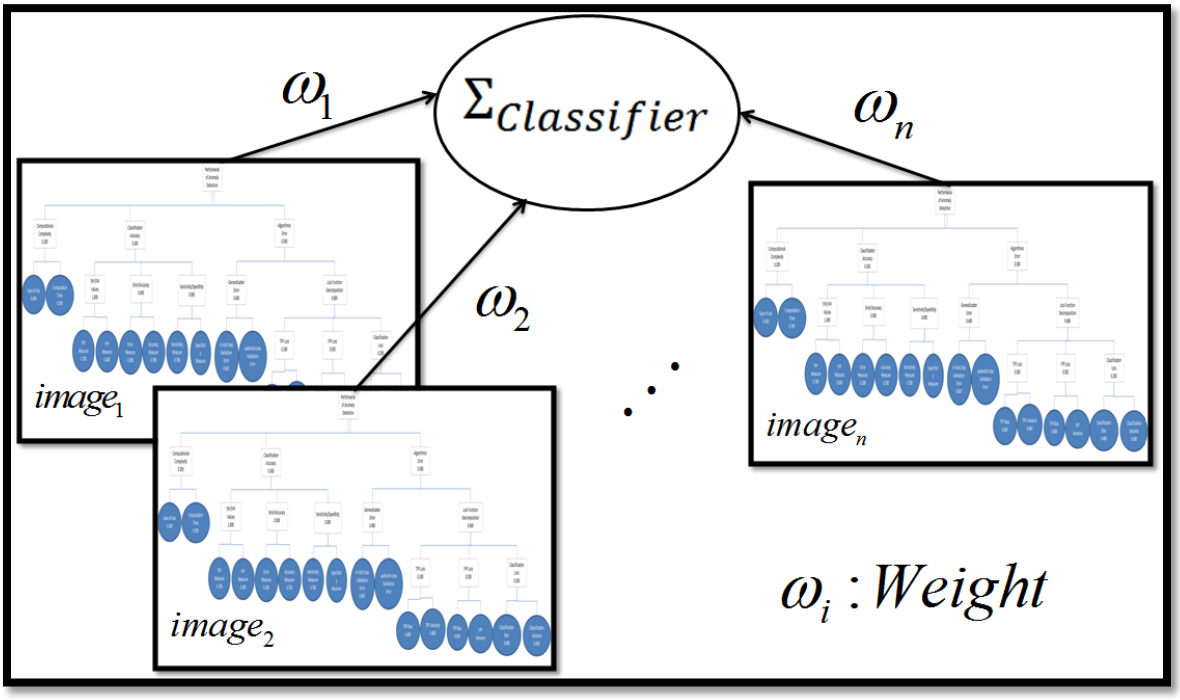


Figure 49. - VFT Image Weighting Process

A quicker way to accomplish this process is to take the medians of the values for certain factor combinations. This decreases the size of the problem, as it folds over on the images and only provides an output for a certain factor combination. In this study, the target pixel percentage and Mahalanobis distance levels were used to generate median image values and then were put through the hierarchy.

Step 3: Developing Evaluation Measures

The SME had told me to rank the TPF Measure higher than the FPF measure, but the sensitivity of the FPF measure and the accuracy and precision of its calculation was of a higher concern. This is why, paradoxically, the TPF Measure is weighted larger than the FPF measure, but the FPF Bias and Variance are weighted larger than the TPF Bias and Variance. This makes sense in the HSI setting as groups of pixels that are labeled falsely as targets will only appear in certain images taken of the same area and thus, there is an inherent bias/variance of the existence of a False Positive in the image. By keeping this value as low as possible, the FPF measure should also be held as low as possible.

Step 4: Creating Value Functions

Value functions were created with the use of SME input. These are shown below as well as in Appendix A.

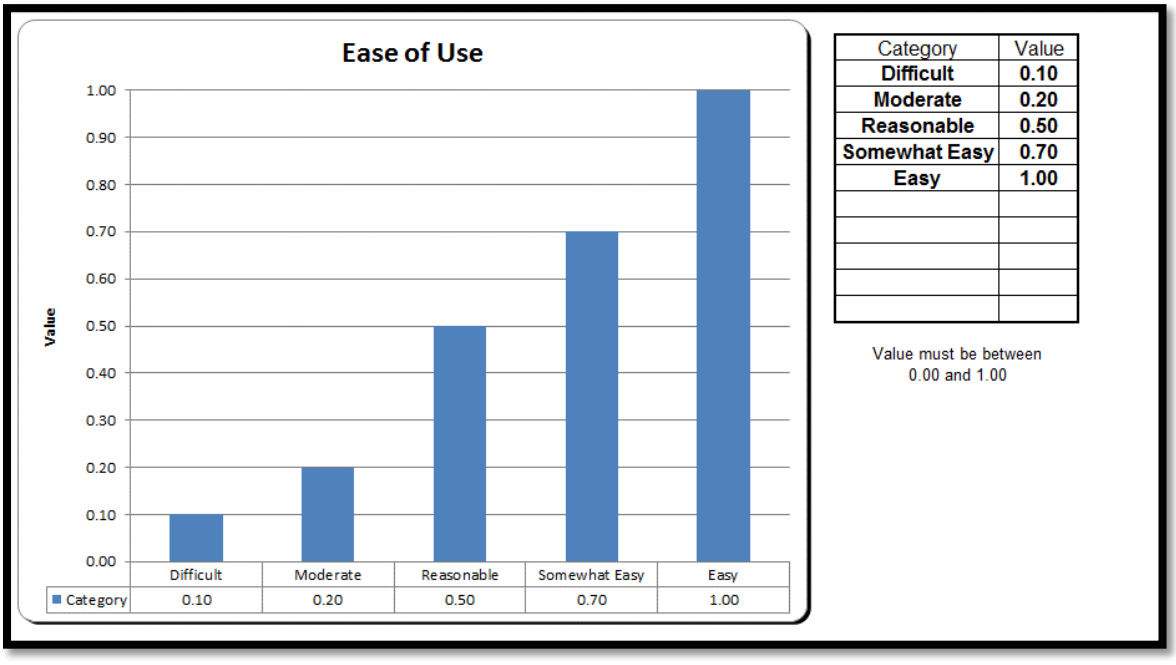


Figure 50. - Ease of Use Value Function

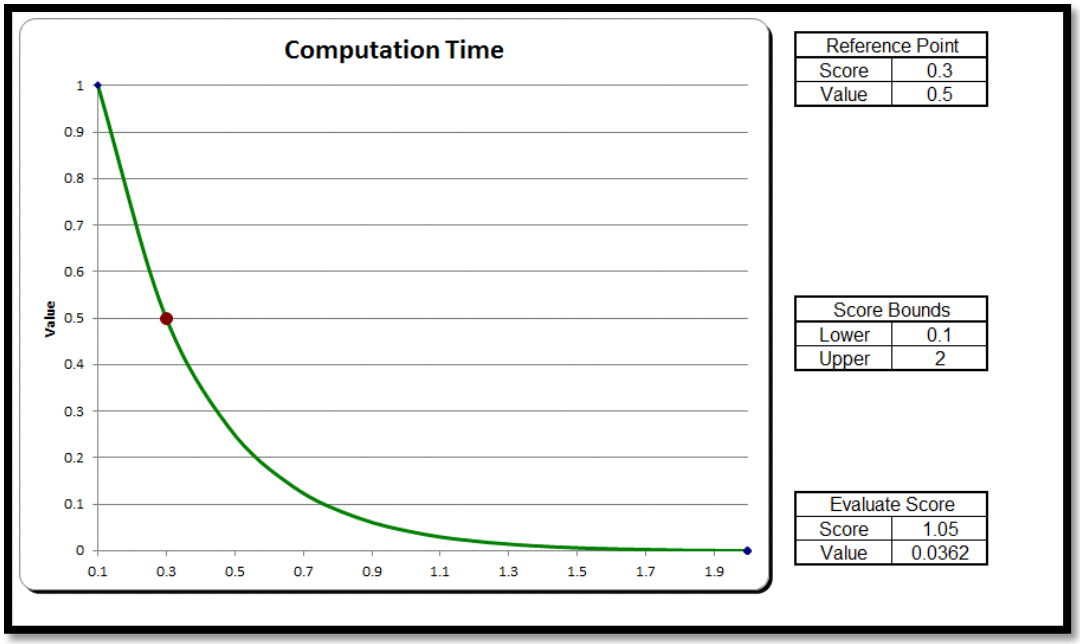


Figure 51. - Computation Time Value Function

Step 5: Weighting the Value Hierarchy

The hierarchy was weighted both globally and locally using a by branch and tier weighting that allowed the decision maker to provide input to what value or measure was more important across only a few values and measures. Figures 52 to 57 show the Global and Local weightings for each branch of the hierarchy.

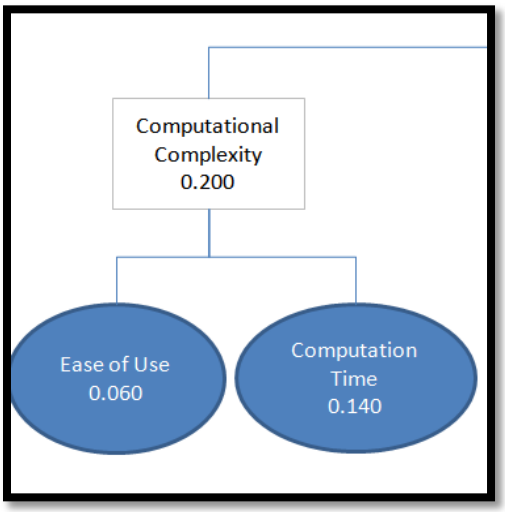


Figure 52. - Global Weights - Computational Complexity

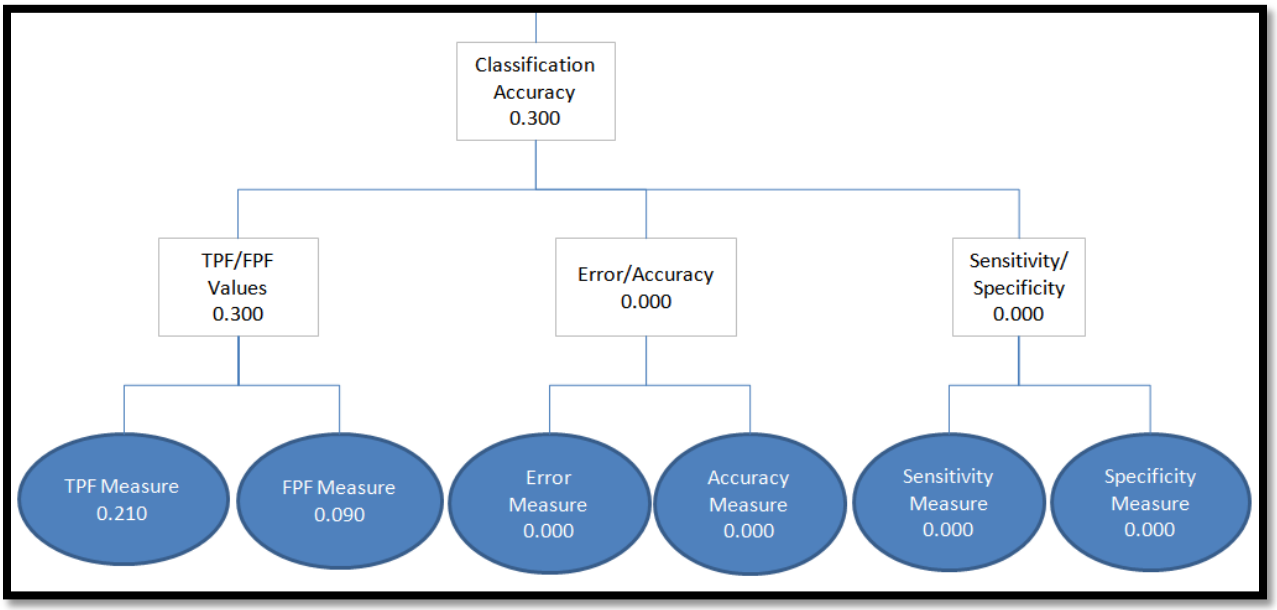


Figure 53. - Global Weights - Classification Accuracy

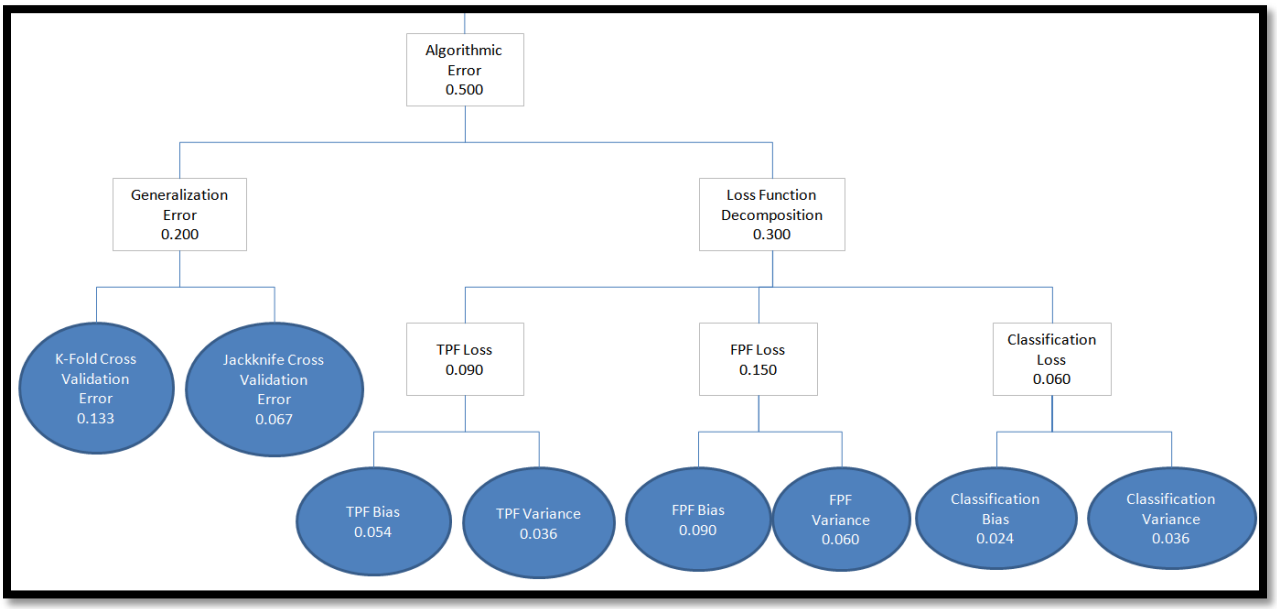


Figure 54. - Global Weights - Algorithmic Error

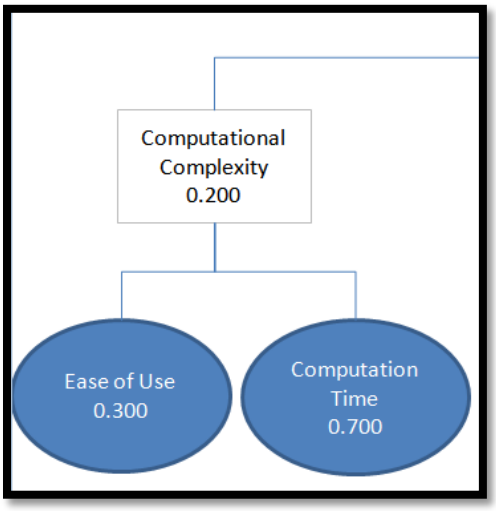


Figure 55. - Local Weights - Computational Complexity

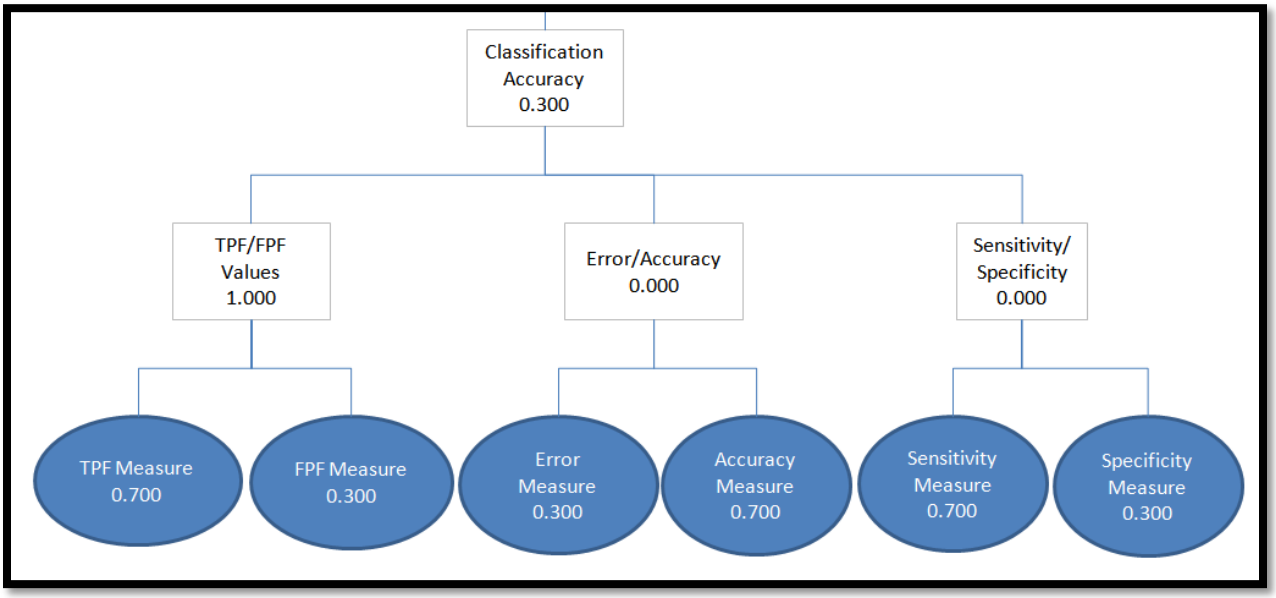


Figure 56. - Local Weights - Classification Accuracy

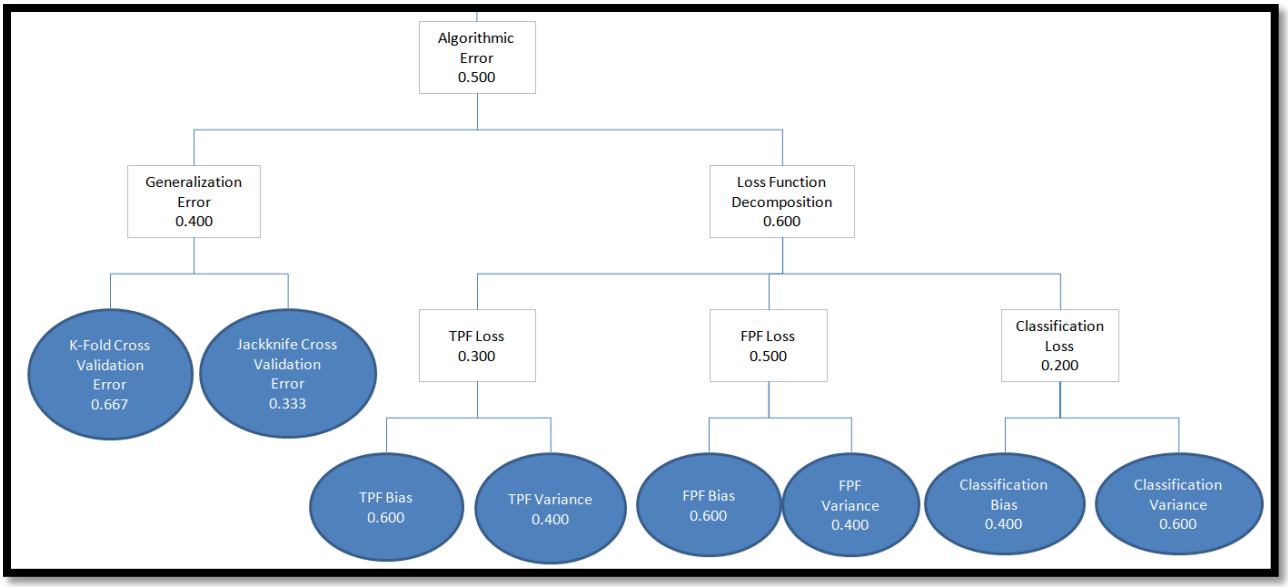


Figure 57. - Local Weights - Algorithmic Error

These weights were then ordered and analyzed to understand if any weights were out of proportion with the other values or measures in terms of importance. These weights would then need to be reweighed which is not a difficult process. The weights that were determined in this case were reflective of the importance of all of the variables involved. These can be seen below. The following table features the local and global weights for the values. Global weights can be derived from local weights by multiplying down branches.

Table 16. - Local and Global Weights for Values

Value	Tier	Local Weight	Global Weight
Performance	1	1	1
Computational Complexity	2	0.2	0.2
Classification Accuracy	2	0.3	0.3
TPF/FPF Values	3	0.3	0.3
Error/Accuracy	3	0	0
Sensitivity/Specificity	3	0	0
Algorithmic Error	2	0.5	0.5
Generalization Error	3	0.4	0.2
Loss Function Decomposition	3	0.6	0.3
TPF Loss	4	0.3	0.09
FPF Loss	4	0.5	0.15
Classification Loss	4	0.2	0.06

The measures were also weighted with the same local and global weighting procedure. The most weight has been placed on the TPF measure while the least weight is placed on the classification bias and variance, as they are fairly new procedures with a large amount of uncertainty. The weights can be changed as more insight is developed.

Table 17. - Local and Global Measure Weights

Measure	Local Weight	Global Weight
Ease of Use	0.3	0.06
Computation Time	0.7	0.14
TPF Measure	0.7	0.21
FPF Measure	0.3	0.09
Error Measure	0.3	0
Accuracy Measure	0.7	0
Sensitivity Measure	0.7	0
Specificity Measure	0.3	0
K-Fold Cross Validation Error	0.667	0.133
Jackknife Cross Validation Error	0.333	0.067
TPF Bias	0.6	0.054
TPF Variance	0.4	0.036
FPF Bias	0.6	0.09
FPF Variance	0.4	0.06
Classification Bias	0.4	0.024
Classification Variance	0.6	0.036

The following table shows the global weights and their rank per the tier for the values. This table is in order of overall global weight. It shows the fact that algorithmic error is weighted the highest amongst the three main branches.

Table 18. - Global Tier Rankings

Value	Tier	Global Weight	Rank (Tier)
Performance	1	1	1
Algorithmic Error	2	0.5	1
Classification Accuracy	2	0.3	2
TPF/FPF Values	3	0.3	1
Loss Function Decomposition	3	0.3	2
Computational Complexity	2	0.2	3
Generalization Error	3	0.2	3
FPF Loss	4	0.15	1
TPF Loss	4	0.09	2
Classification Loss	4	0.06	3
Error/Accuracy	3	0	NA
Sensitivity/Specificity	3	0	NA

The next table shows the ranks of the measures in terms of global weights. The TPF measure is weighted the highest as it is representative of the main goal of target detection, that of actually detecting targets when the pixels are indeed targets. The classification variance and bias are ranked on the bottom, as they are experimental techniques that need to be assessed. The color scheme is seen in the following table.

Table 19. - Global Measure Rankings

Measure	Global Weight	Rank
TPF Measure	0.21	1
Computation Time	0.14	2
K-Fold Cross Validation Error	0.133	3
FPF Measure	0.09	4
FPF Bias	0.09	4
Jackknife Cross Validation Error	0.067	6
Ease of Use	0.06	7
FPF Variance	0.06	7
TPF Bias	0.054	9
TPF Variance	0.036	10
Classification Variance	0.036	10
Classification Bias	0.024	12
Error Measure	0	NA
Accuracy Measure	0	NA
Sensitivity Measure	0	NA
Specificity Measure	0	NA

Table 20. - Color Representation in Tables

Value	Color
Computational Complexity	Blue
Classification Accuracy	Green
Algorithmic Error	Yellow

Step 6: Generating Alternatives

As previously covered, alternative classification algorithms, QDA, Naïve Bayes, and CART were chosen to be representative of the whole of pattern classification research. Other algorithms could be chosen with different parameters as needed.

Step 7: Scoring Alternatives

Alternatives were scored using the measures and methods seen in the previous methodology chapter. In this case, the median of the factor combinations were used to assess the alternative score. Various different tactical decisions had to be made when scoring the alternatives and generating the actual value for the measures. These included removing certain

data runs due to their inability to be classified well by the classifiers. Some of the points with large Mahalanobis distances are uninteresting problems and can be discarded as any of the classifiers can arbitrarily classify them well. This idea can be seen the following matrix plot of the biases and variances that were generated. As the Mahalanobis distance increased, the bias and variance shrinks to zero, as the classifier will always classify the problem well. There is a cutoff at around 10 where the bias and variance reach a level that is effectively small.

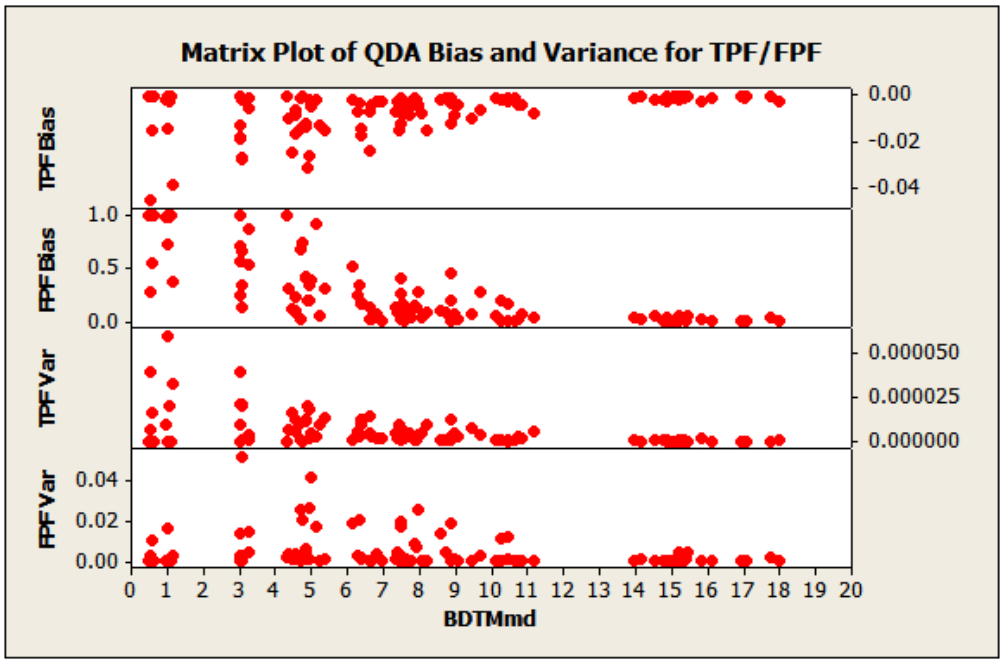


Figure 58. - TPF/FPF Bias/Variance vs. Mahalanobis Distance

Additionally, when assessing the Misclassification Rate across various folds of the k-fold Cross Validation methodology, it was of interest to see which fold number would result in the lowest MCR rate. These MCR rates were calculated for different target pixel percentage levels (1%, 5%, 10%), which was a significant factor. The 1% level resulted in the lowest MCR rate. It was determined that there was no significant difference across the folds, and the lowest amount of folds necessary can be used in this scenario. The default of 10 could also be used.

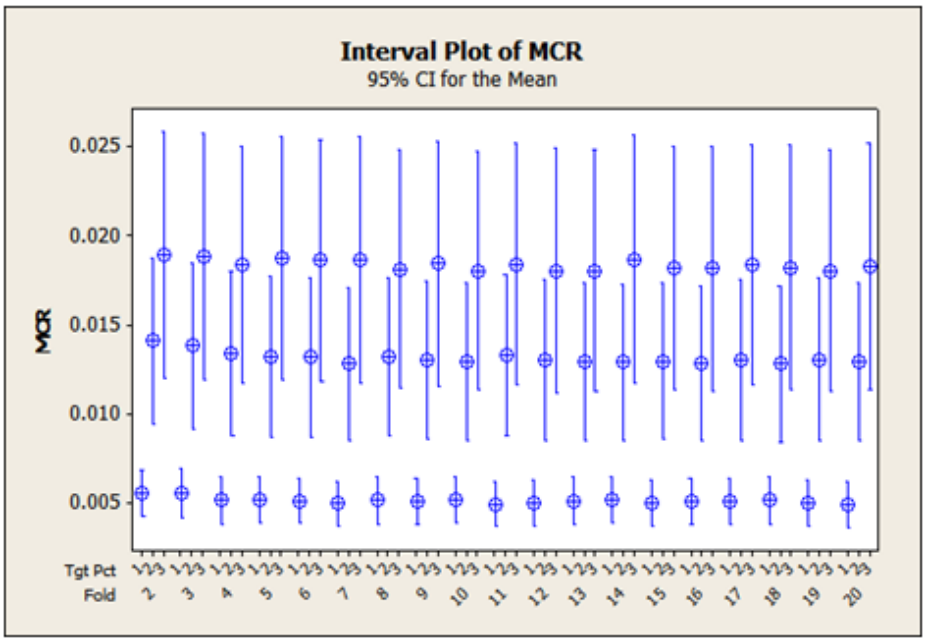


Figure 59. - Misclassification Rate vs. Target Pixel Percentage and Fold Number

An assessment of the jackknife MCR was also accomplished for the three different classifiers. Figure 60 shows the MCR rates for each image from 0 to 243. It was seen that as the number of target pixels increased, the MCR rate also increased. This happened across all of the algorithms.

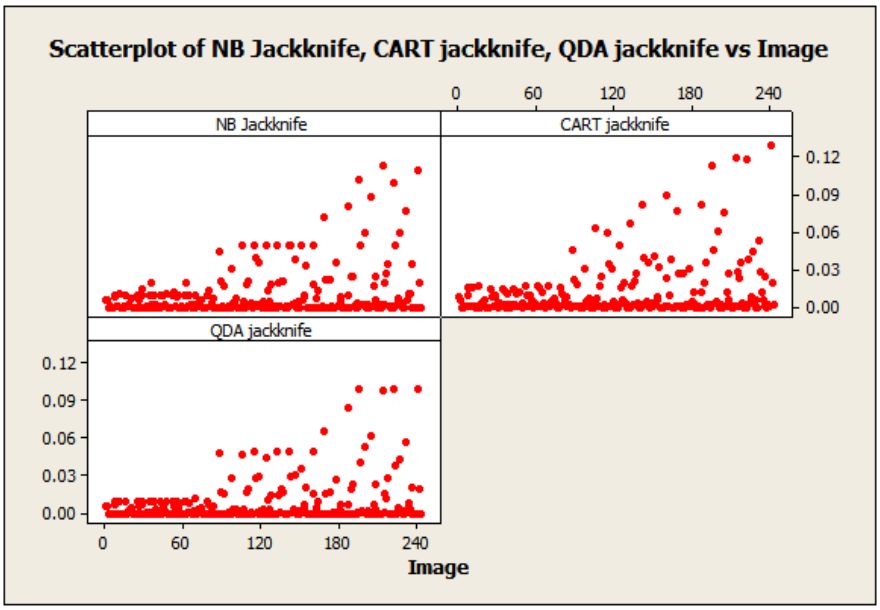


Figure 60. - Jackknife Misclassification Rates

Other tactical decisions that had to be made were the size of the tree and where to prune the tree and the methodology to prune the tree in order to avoid over-fitting. This was assessed for MCR at level of trees and at minimum leaf node sizes.

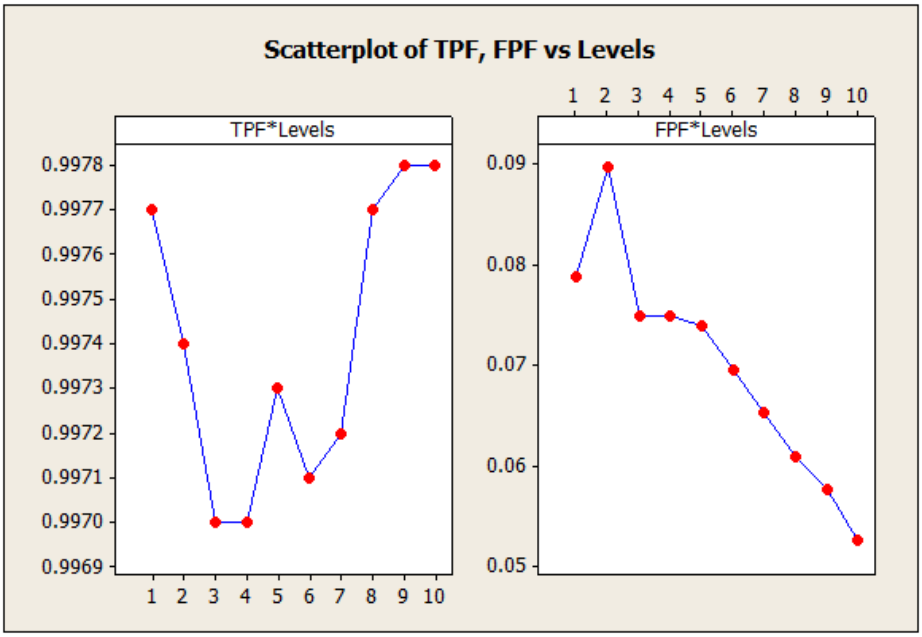


Figure 61. - TPF, FPF vs Pruning Level

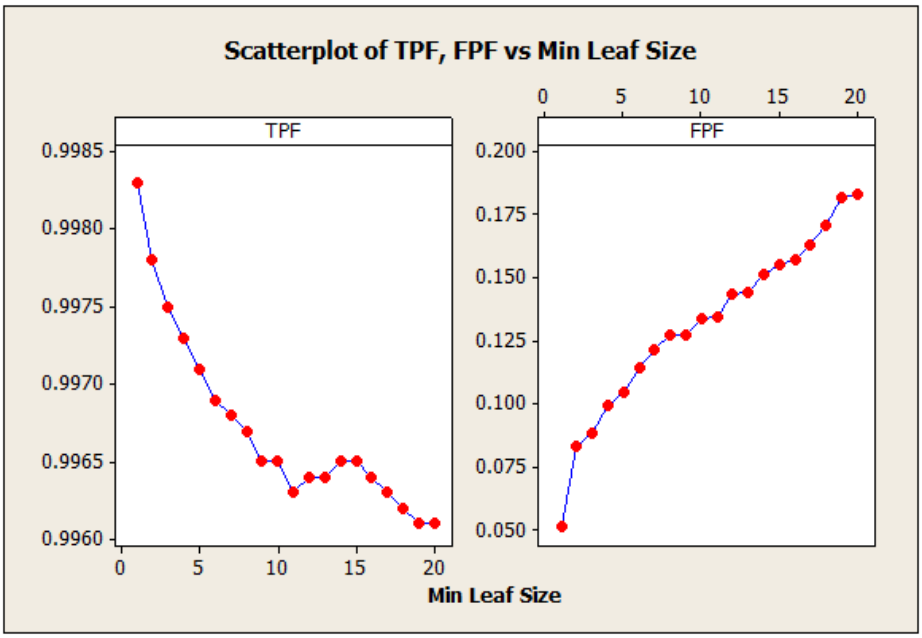


Figure 62. - TPF, FPF vs. Leaf Size

A pruning level of five was chosen to be adequate to avoid over-fitting.

In order to collect the biases and variances using Domingos' formulation, a parametric bootstrapping technique was accomplished. This fit the overall run distributions with the parametric classifier first and then this fit was tested with random generations of data from the same distributions that were used in the first fit. The fit using these random points, which represent different instances of sample reality from an underlying true population, were then tested on a grid of points. This idea is seen below.

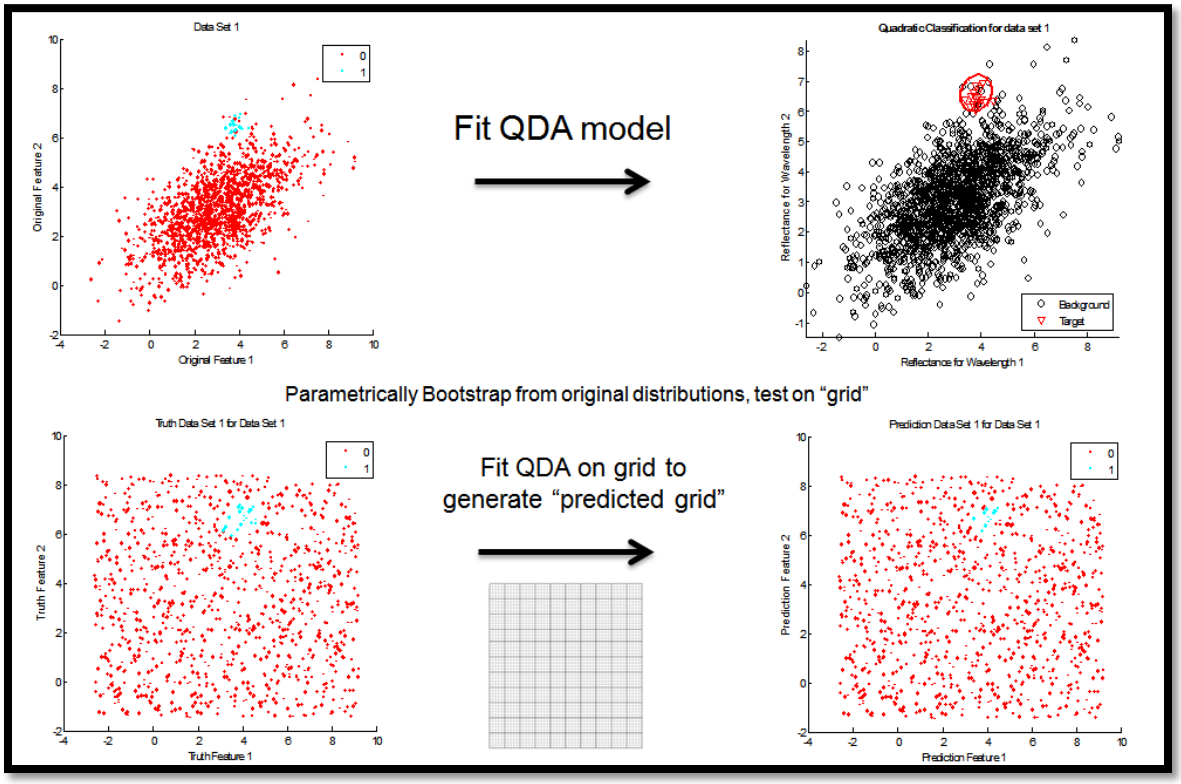


Figure 63. - Domingos Bias/Variance Methodology

New fits of these sample truths were then created and the differences for each of these data points from each truth was used to determine bias and variance. This was then integrated across all points in the grid in order to get a result for the bias and variance.

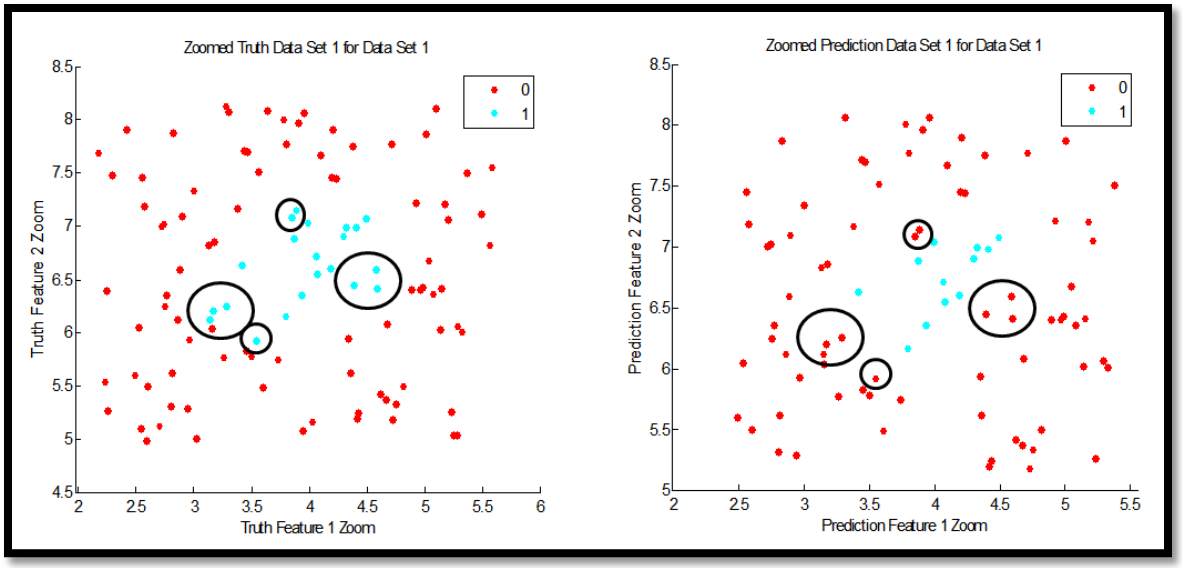


Figure 64. - Domingos' Boundary Error

The overall scoring of the alternatives for the computational complexity measures is seen in Table 21 and Figure 65. CART had the fastest times while QDA had the slowest.

Table 21. - Computational Complexity Measures

Algorithm	Mdist	Pix Pct	Ease of Use	Computation Time
QDA	Long	1%	Somewhat Easy	0.017981996
CART	Long	1%	Reasonable	0.001900333
NB	Long	1%	Somewhat Easy	0.011519248
QDA	Short	1%	Somewhat Easy	0.017981601
CART	Short	1%	Reasonable	0.001904873
NB	Short	1%	Somewhat Easy	0.012695926
QDA	Long	5%	Somewhat Easy	0.017977259
CART	Long	5%	Reasonable	0.00190507
NB	Long	5%	Somewhat Easy	0.011639659
QDA	Short	5%	Somewhat Easy	0.017976469
CART	Short	5%	Reasonable	0.001906452
NB	Short	5%	Somewhat Easy	0.014577308
QDA	Long	10%	Somewhat Easy	0.01798476
CART	Long	10%	Reasonable	0.001911979
NB	Long	10%	Somewhat Easy	0.011722171
QDA	Short	10%	Somewhat Easy	0.017996998
CART	Short	10%	Reasonable	0.001919875
NB	Short	10%	Somewhat Easy	0.014997959

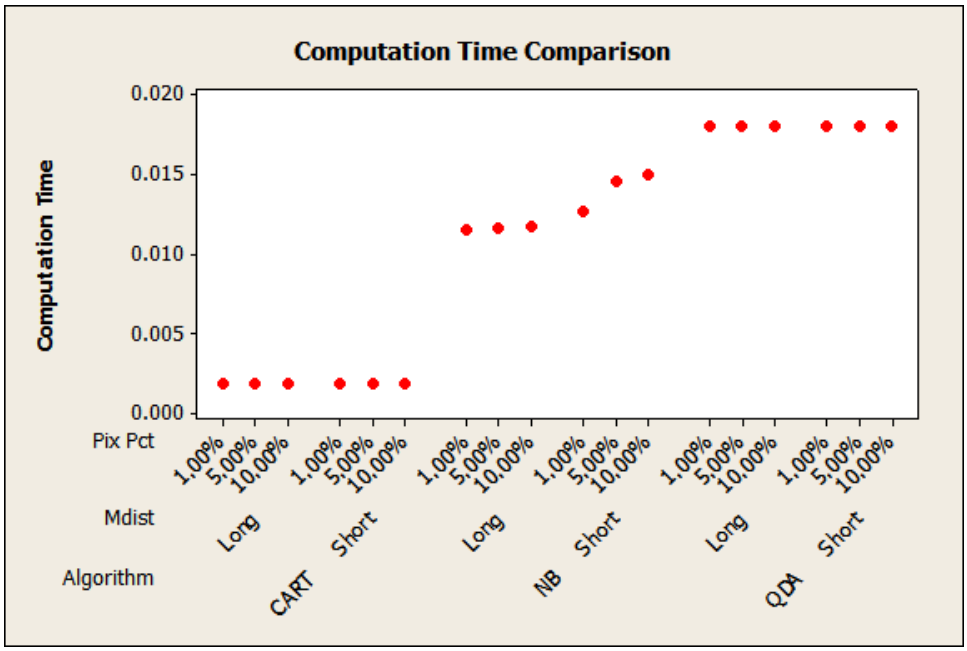


Figure 65. - Computation Time Comparion

The values for the measures for classification are shown in Table 22 and a comparison of FPF Measures is shown in Figure 66. For the Long level of the Mahalanobis Distance factor, each algorithm resulted in 0 FPF. For the Short level, QDA outperformed CART at the 1% and 10% levels. Naïve Bayes resulted in an improvement at the 1% and 5% levels.

Table 22. - Classification Accuracy Measures

Algorithm	Mdist	Pix Pct	TPF Measure	FPF Measure	Error Measure	Accuracy Measure	Sensitivity Measure	Specificity Measure
QDA	Long	1%	1	0	1	0	1	1
CART	Long	1%	1	0	1	0	1	1
NB	Long	1%	1	0	1	0	1	1
QDA	Short	1%	0.9971591	0.3125	0.995	0.005	0.99715909	0.6875
CART	Short	1%	0.9971591	0.40625	0.9934375	0.0065625	0.99715909	0.59375
NB	Short	1%	0.9993687	0.21875	0.996875	0.003125	0.99936869	0.78125
QDA	Long	5%	1	0	1	0	1	1
CART	Long	5%	1	0	1	0	1	1
NB	Long	5%	1	0	1	0	1	1
QDA	Short	5%	0.9888158	0.15	0.98375	0.01625	0.98881579	0.85
CART	Short	5%	0.9914474	0.15	0.98	0.02	0.99144737	0.85
NB	Short	5%	0.9953947	0.0625	0.986875	0.013125	0.99539474	0.9375
QDA	Long	10%	1	0	1	0	1	1
CART	Long	10%	1	0	1	0	1	1
NB	Long	10%	1	0	1	0	1	1
QDA	Short	10%	0.9871528	0.09375	0.978125	0.021875	0.98715278	0.90625
CART	Short	10%	0.9829861	0.1125	0.9746875	0.0253125	0.98298611	0.8875
NB	Short	10%	0.9892361	0.096875	0.97625	0.02375	0.98923611	0.903125

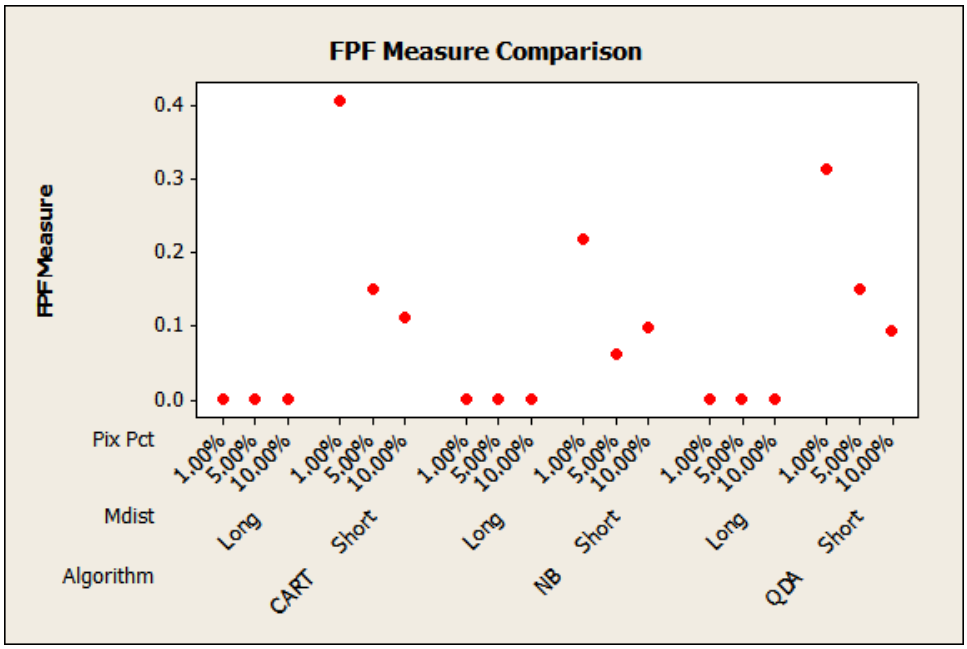


Figure 66. - FPF Measure Comparison

The algorithmic error measure values are shown in Table 23 and the K-fold and Jackknife Cross-Validation error measures are shown in Figures 67 and 68 respectively. CART outperformed both NB and QDA in both measures. Jackknife resulted in slightly less error than k-fold.

Table 23. - Algorithmic Error Measures

Algorithm	Mdist	Pix Pct	K-fold Cross Validation Error	Jackknife Cross Validation Error	TPF Bias	TPF Variance	FPF Bias	FPF Variance	Domingos Classification Bias	Domingos Classification Variance
QDA	Long	1%	0	0	0	0	0	0	0.014	0.00381
CART	Long	1%	0.00125	0.00125	0	0	0.006142	0.0000729	0.016	0.00322
NB	Long	1%	0.0009375	0.000625	-6.3E-07	3.94E-10	0.00025	0.0001249	0	0.00926
QDA	Short	1%	0.0065625	0.00625	-0.0024	2.05E-06	0.277979	0.0160991	0.01	0.00234
CART	Short	1%	0.0034375	0.00375	-0.00035	5.175E-07	0.063544	0.0014403	0.0085	0.00237
NB	Short	1%	0.0090625	0.00875	-0.00082	6.71E-07	0.100434	0.008792	0	0.00133
QDA	Long	5%	0	0	0	0	0	0	0.016	0.00443
CART	Long	5%	0.00125	0.00125	0	0	0.006348	0.0000289	0.02	0.00323
NB	Long	5%	0.00125	0.00125	0	0	0.000319	6.79E-06	0	0.01299
QDA	Short	5%	0.0175	0.016875	-0.0106	8.47E-06	0.146768	0.0015322	0.016	0.00358
CART	Short	5%	0.00125	0.001875	0	0	0.082656	0.000393	0.014	0.00168
NB	Short	5%	0.025625	0.0225	-0.0084	0.0000303	0.098892	0.0062064	0.001	0.00852
QDA	Long	10%	0	0	0	0	0	0	0.016	0.004
CART	Long	10%	0.0025	0.0025	-0.0011	7.225E-07	0.011727	0.0000663	0.02	0.00418
NB	Long	10%	0.0009375	0.00125	0	0	0.00071	7.26E-06	0	0.01425
QDA	Short	10%	0.021875	0.021875	-0.01242	9.26E-06	0.093111	0.0004688	0.0195	0.00275
CART	Short	10%	0.00125	0.00125	0	0	0.020887	0.0002702	0.0205	0.00244
NB	Short	10%	0.0321875	0.029375	-0.01281	0.0000556	0.097616	0.0020594	0.0025	0.00073

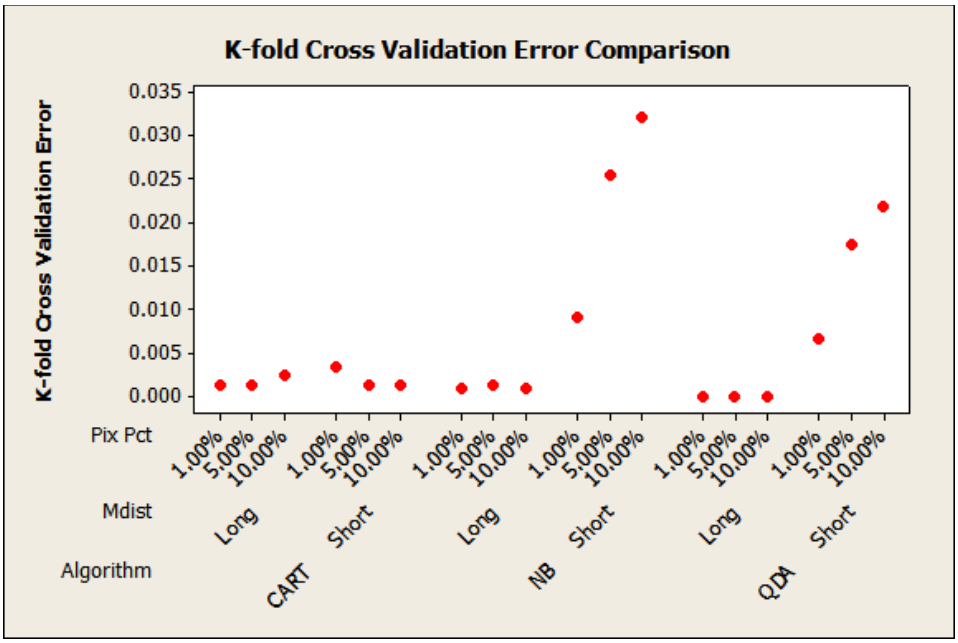


Figure 67. - K-fold Cross Validation Error Comparison

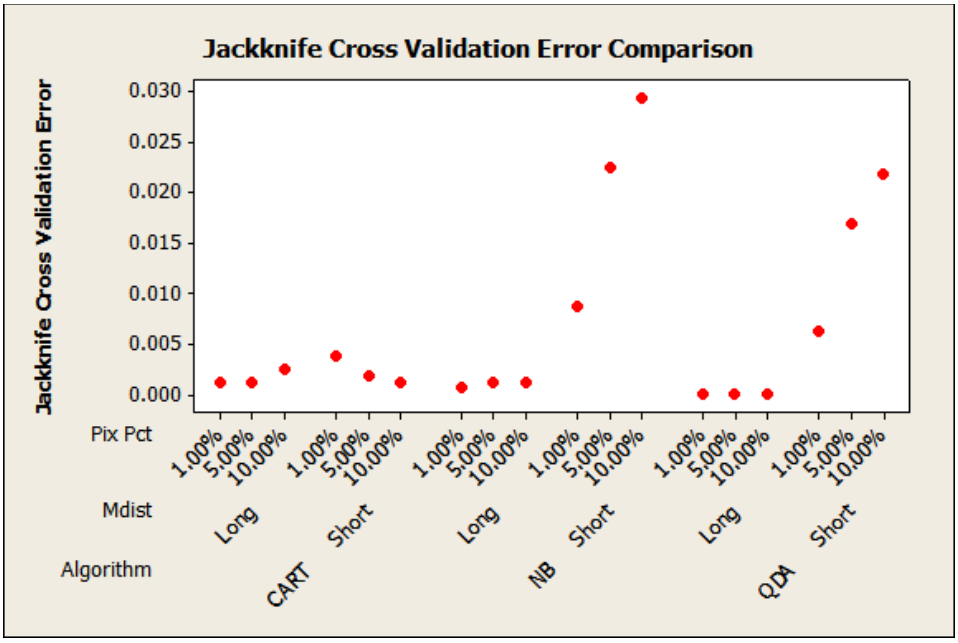


Figure 68. - Jackknife Cross Validation Error Comparison

TPF Bias/Variance comparisons are shown in Figure 69 and FPF Bias/Variance comparisons are shown in Figure 70. Again, values are split for both Mahalanobis distance and target percentage factors. CART performs well in each of these measures. QDA performs well for TPF Variance but maintains some bias. NB is shown to have variance at the 5% and 10% levels and also maintains bias at these levels. The variance values are shown to be very small and could be operationally insignificant.

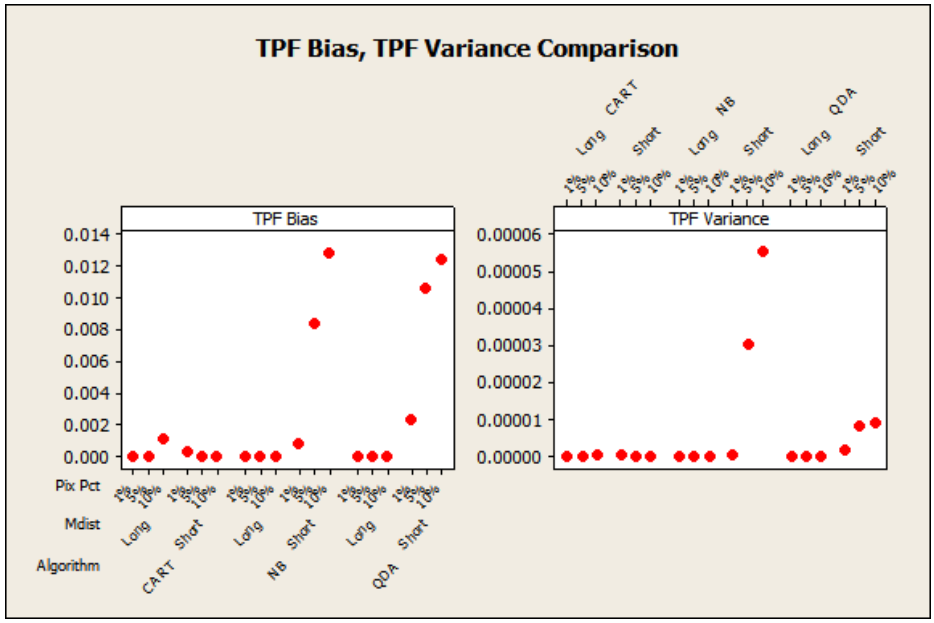


Figure 69. - TPF Bias/Variance Comparison

For FPF Bias and Variance, NB has a consistent bias across each short level at around 0.1 while QDA has a relatively large bias at the 1% target pixel percentage level. QDA performs well for 5% and 10% but has a large variance for 1%. This could be due to the QDA algorithm predicting close to all positives when there are less target pixels present. CART still is shown as the best option in these cases. The FPF Variance is larger in value than the TPF Variance.

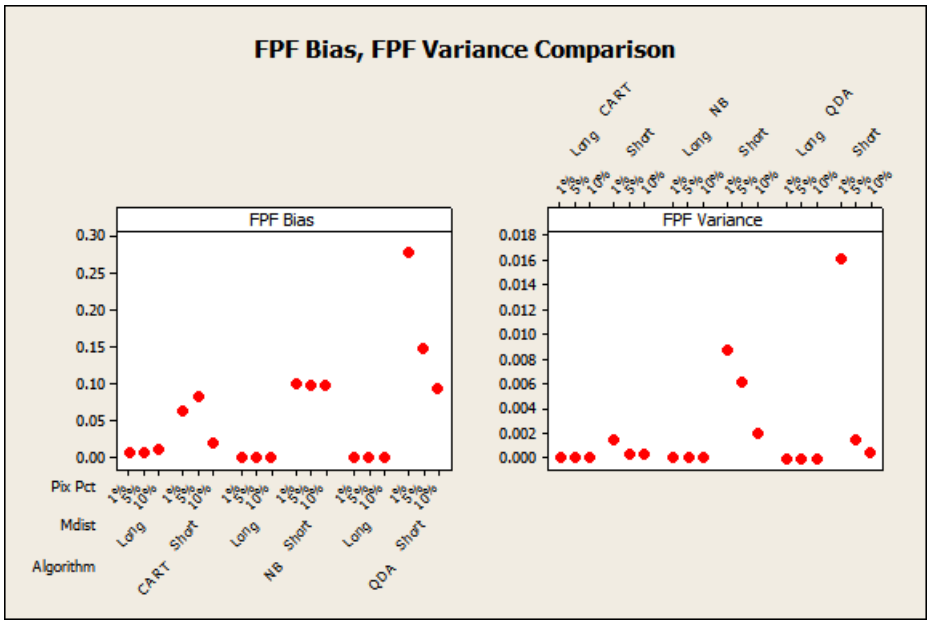


Figure 70. - FPF Bias/Variance Comparison

The Domingos' Bias/Variance comparison is shown in Figure 71. This methodology shows that NB is the best performing algorithm and CART now is shown to have bias. Interestingly, the long Mahalanobis distances are now registering positive biases and variances, which could be due to the non-parametric bootstrap approach. Also, CART performs the worst in this situation in terms of bias, which could mean that each tree that is being built is resulting in different decisions for the border pixels. NB has more variance than the other two algorithms, even though it has the least bias. This seems to show that NB is adept to fitting fairly accurate decision boundaries, but they change from situation to situation at a higher rate than the other algorithms. The Domingos' variance is similar to the variance observed for the FPF values.

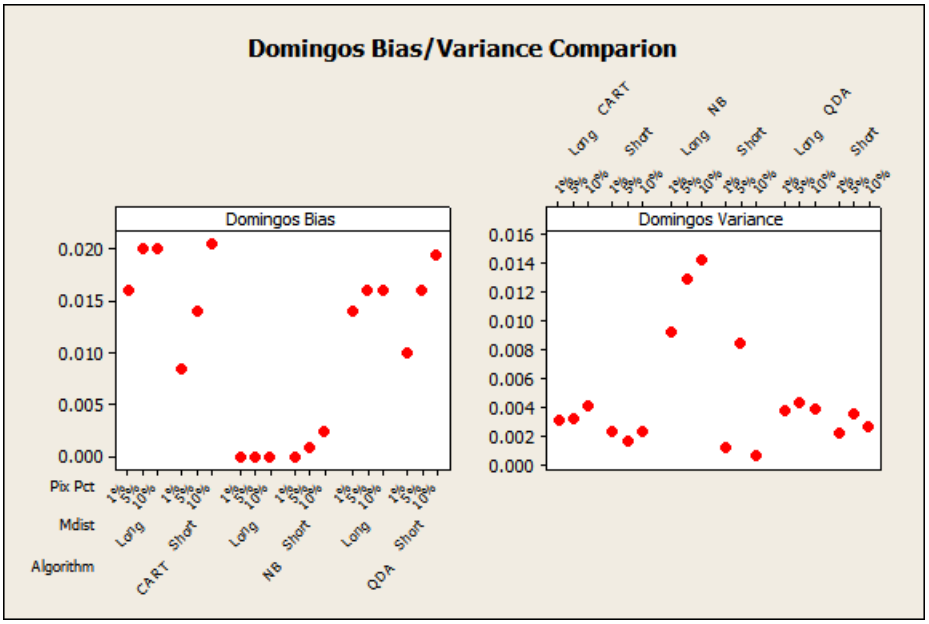


Figure 71. - Domingos' Bias/Variance Comparison

A listing of results for the comparison of Domingos' Bias/Variance values to TPF and FPF bias and variance values for all image cases are shown in Appendix B. FPF bias is always the highest amount of bias observed for each algorithm. This is due to some runs being classified as all background when the target percentage or Mahalanobis distance is small, which results in higher FPF values. The Domingos' variance values seem to be larger than either TPF or FPF values when accounting for all of the runs, which is not apparent when simply looking at the median values.

Step 8: Deterministic Analysis

All of the algorithms were ranked per the pixel percentage and the Mahalanobis distances. It was seen that CART always outperforms the other algorithms in each situation, although there is a dropoff in performance from long to short distance. QDA is always the worst performing algorithm, primarily due to its generalizability capability (although in all of the examples, the distributions had positive covariances). The analysis in the previous step showed

why QDA performed at a lower level than the other algorithms, especially in computation time, FPF Variance, and TPF Bias.

Table 24. - Aggregated Hierarchy Values

Algorithm	Mdist	Pix Pct	Hierarchy
QDA	Long	1%	0.848
CART	Long	1%	0.919
NB	Long	1%	0.87
QDA	Short	1%	0.57
CART	Short	1%	0.756
NB	Short	1%	0.662
QDA	Long	5%	0.843
CART	Long	5%	0.915
NB	Long	5%	0.864
QDA	Short	5%	0.537
CART	Short	5%	0.738
NB	Short	5%	0.591
QDA	Long	10%	0.845
CART	Long	10%	0.903
NB	Long	10%	0.863
QDA	Short	10%	0.56
CART	Short	10%	0.72
NB	Short	10%	0.564

Figure 72 shows the significant difference between long and short Mahalanobis distance levels and the fact that CART is outperforming the other two algorithms in each case.

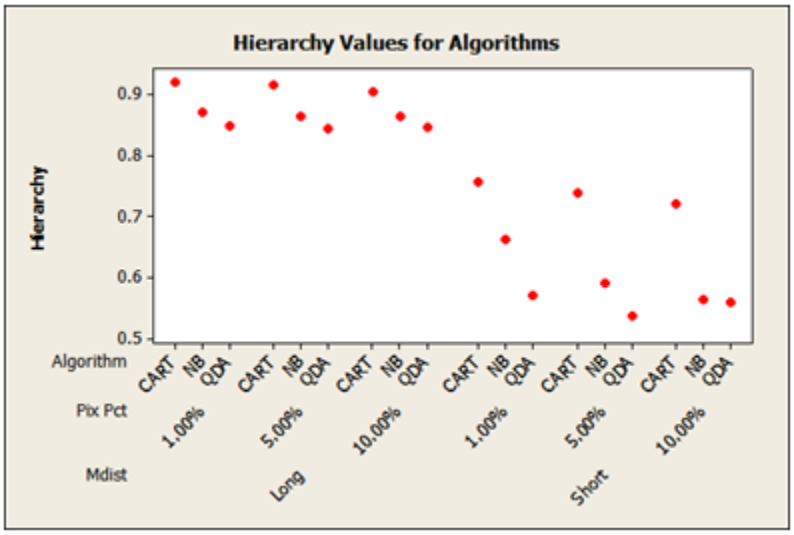


Figure 72. - Hierarchy Values per Target Pixel Percentage and Mahalanobis Distance

The following figures are breakdowns of the value quantities for each algorithm in each different factor combination. They are useful for visualizing the areas that the algorithms outperform the others in. For example, for the first case of 1% Target Pixels and Short Mahalanobis distance, CART outperforms the other two algorithms in computation time, while NB outperforms the others in TPF measure. Additionally, QDA performs poorly in this case in terms of FPF bias which is driving the overall hierarchy value down. Also, CART is the only algorithm to show a value for FPF variance, which adds to its quality. Therefore, the largest discriminators are computation time, FPF bias, and FPF variance. If these measures are disproportionally affecting the value of the algorithm, the DM may decide to change the weight or value functions associated with the particular measures.

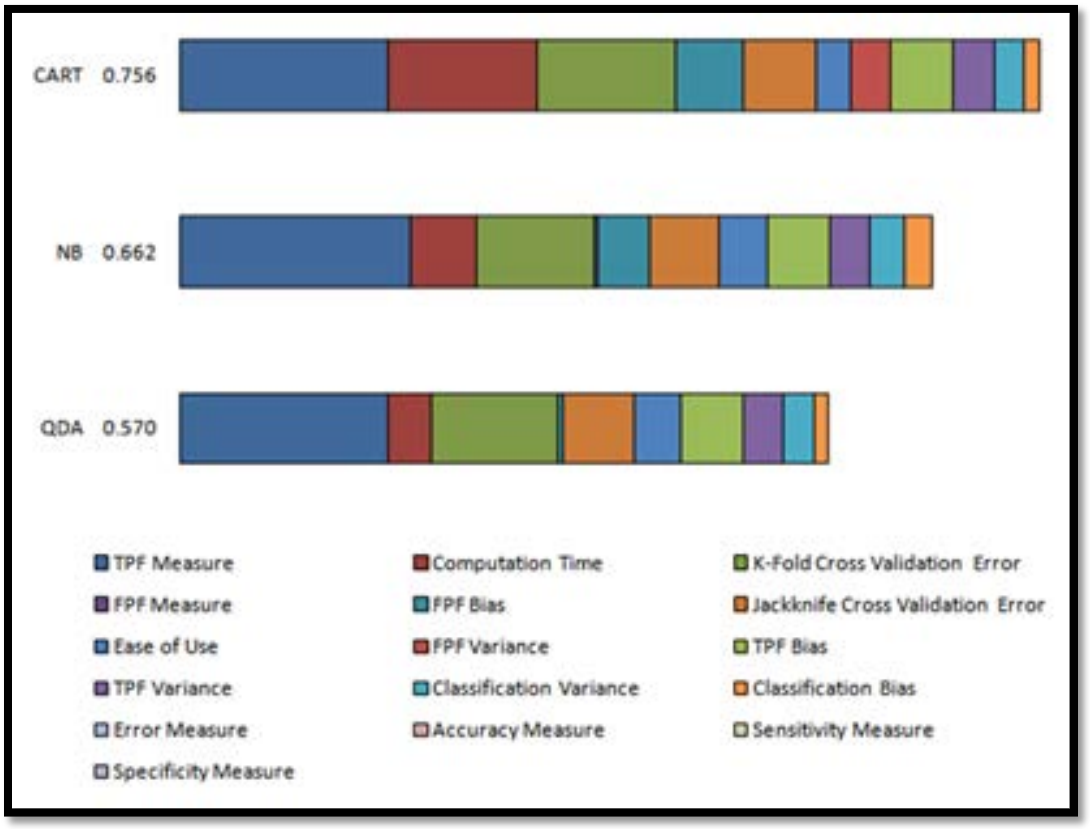


Figure 73. - Hierarchy Values for 1% Target Pixel Pct and Short Mahalanobis Dist

Figures 74 through 78 show a similar breakdown of the value scores for each particular measure.

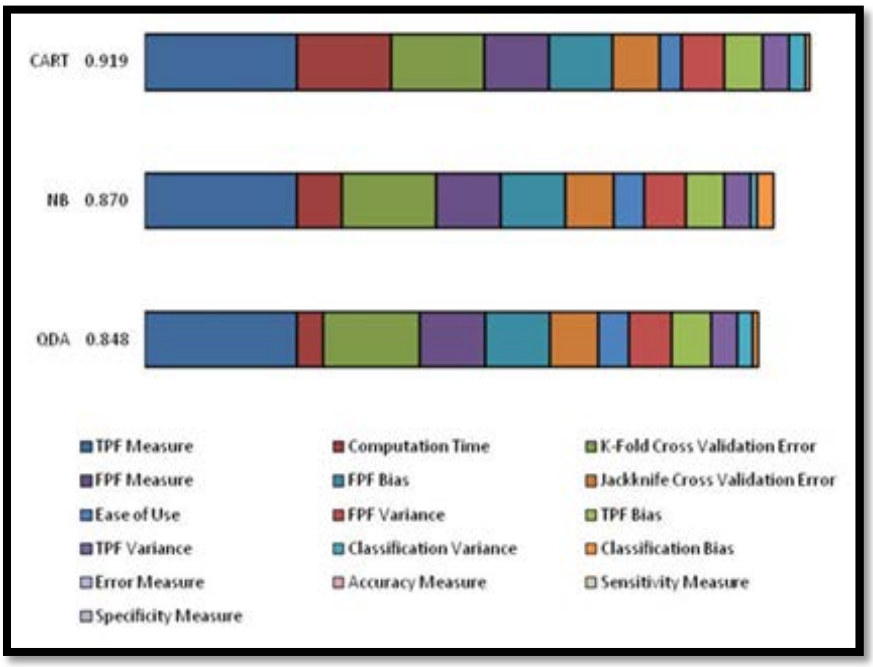


Figure 74. - Hierarchy Values for 1% Target Pixel Pct and Long Mahalanobis Dist

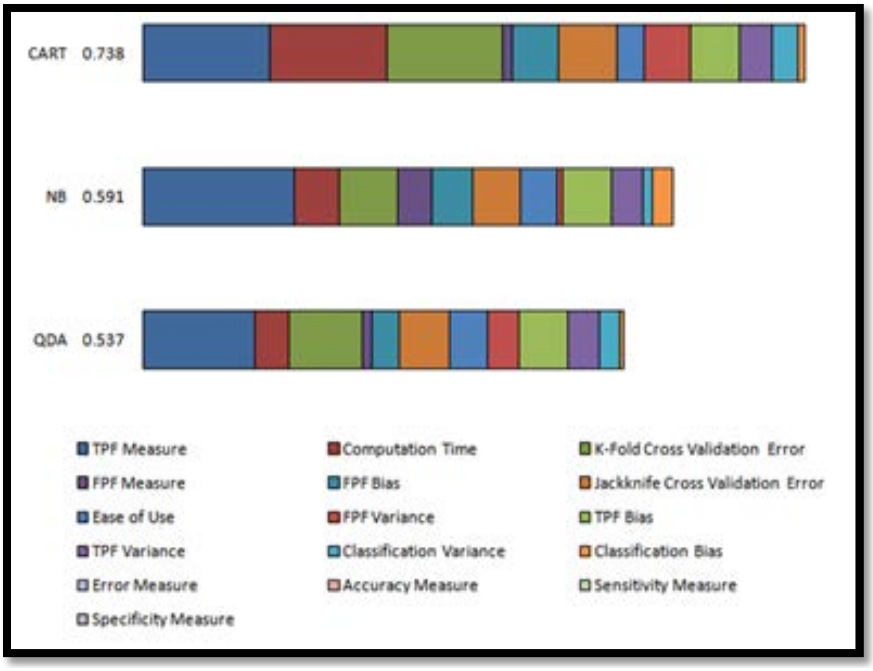


Figure 75. - Hierarchy Values for 5% Target Pixel Pct and Short Mahalanobis Dist

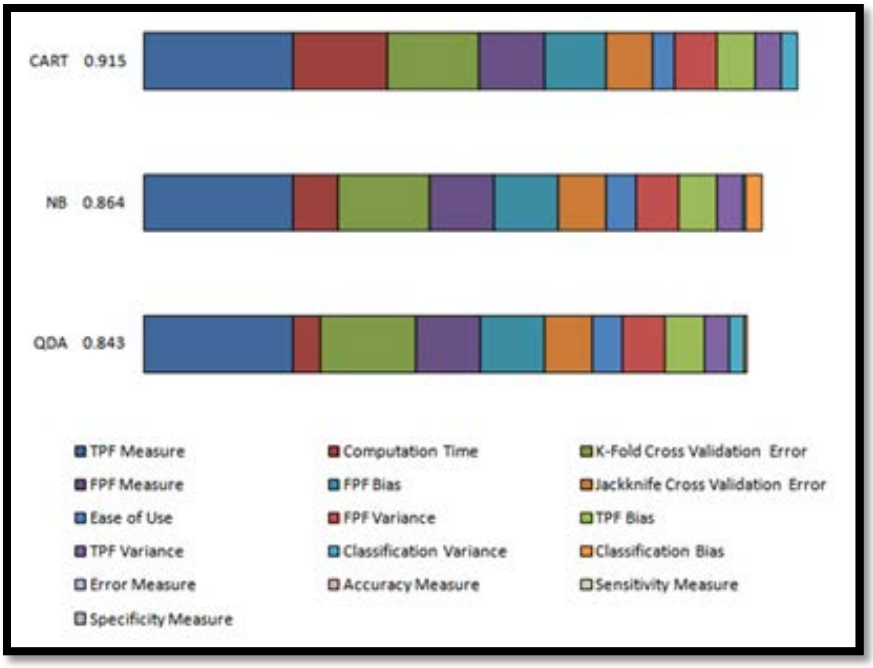


Figure 76. - Hierarchy Values for 5% Target Pixel Pct and Long Mahalanobis Dist

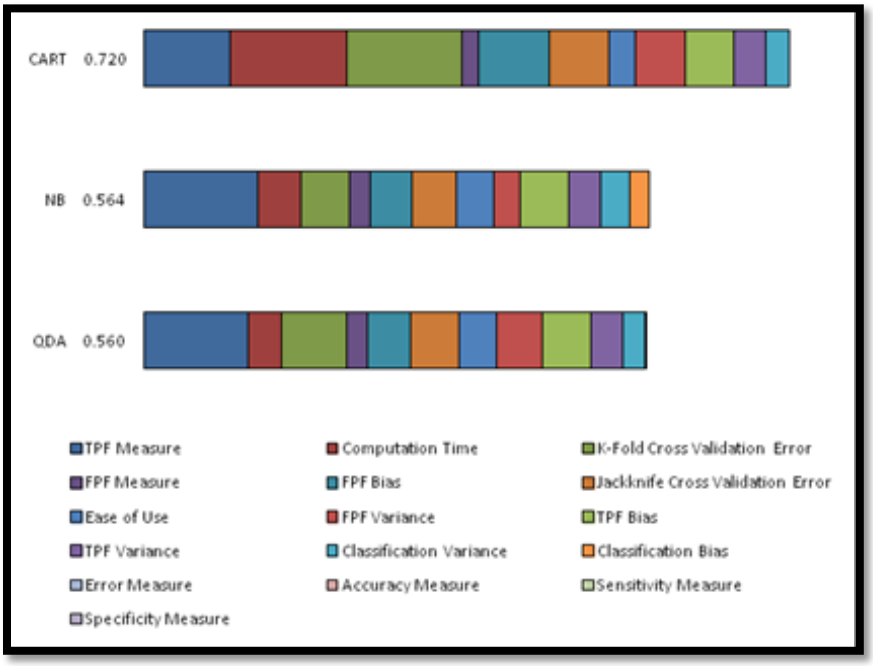


Figure 77. - Hierarchy Values for 10% Target Pixel Pct and Short Mahalanobis Dist

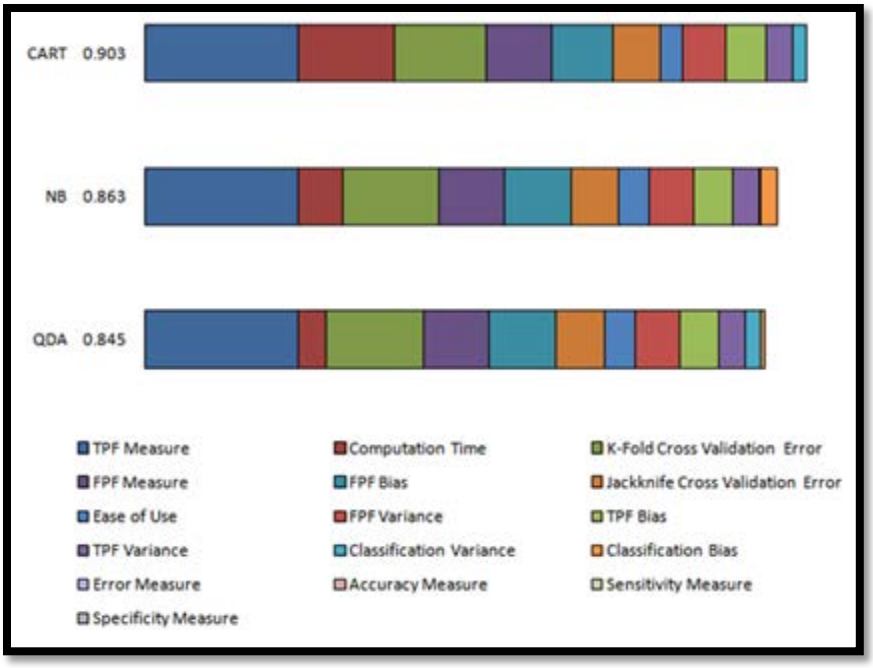


Figure 78. - Hierarchy Values for 10% Target Pixel Pct and Long Mahalanobis Dist

Step 9: Sensitivity Analysis

Sensitivity analysis was accomplished for the algorithms and it was seen that most of the measures were robust. This is seen below as CART is always the best across all ranks.

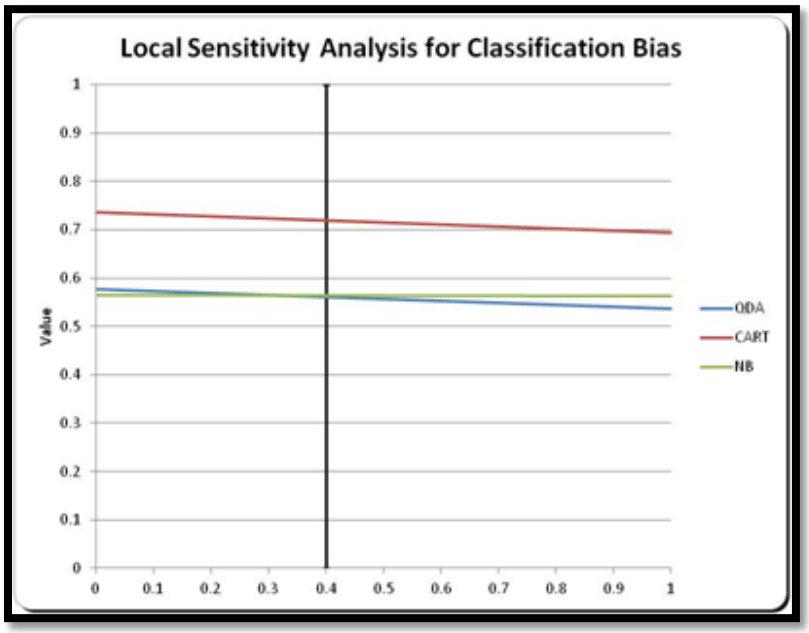


Figure 79. - Classification Bias Local Sensitivity Analysis

Step 10: Conclusions and Recommendations

From this methodology, it is obvious that CART performs the best. However, this is most likely due to the difficulty of the problem and the separation that CART can create between classes.

V. Conclusions

This research is a look into a more robust, transparent, and informative methodology for comparing the performance of pattern classification and hyperspectral imaging algorithms to gain insight about how each algorithm performs under certain problem difficulty and assumptions. The comparison of types of biases and types of cross-validation provides a useful framework for decisions in these areas. The utilization of the value-focused thinking process is an additional benefit that could provide analysts and decision makers a logical and speedy process to combine subjective and objective measurements in deciding which path they may take during a project.

Original Contributions

1. Developed a unique value hierarchy for comparison of different algorithms and carried the analysis through the ten steps to provide an example of how this process can be applied to technical decision making.

2. Provided a comparison of user complexity measures, systematic classification accuracy measures, and algorithmic generalization error decomposition measures to synthesize an overall value based on the inputs of both the decision maker and the performance of these algorithms using parametric and non-parametric bootstrapped estimates.

3. Provided a comparison and analysis of jackknife and k-fold cross-validation performance for training and testing an algorithm on basic two-distribution data sets.

4. Examined the differences between bias and variance estimates for different loss functions, including a quadratic loss function decomposition and a zero-one loss function decomposition and used both results to help inform a decision.

Limitations

There are a few limitations in this research that simplify the problem from what you would see in a real HSI data experiment. The data that was used for this research was a rudimentary representation of true HSI data and each individual problem was not completely representative of the complexities of true data. While the VFT hierarchy could remain in the structure that is exhibited in this research, it could take a willing and talented analyst some time and thought to reform each individual measurement for different algorithms and different types of datasets with various levels of assumptions. The assumptions that were utilized in this research would need to be manipulated and strengthened or loosened for other analytical efforts.

Additionally, within the process of collecting bias and variance estimates, samples were deleted if they did not result in class distributions that had both target and background classes apparent. This was a reflection of the difficulty and level of information contained in the problem. In real empirical samples, this lack of well-shaped data may not be present, and other ways of combating this problem may need to be developed. The formulation for the non-parametric bootstrap was based on treating each unique x vector and corresponding class label as one case and bootstrapping the individual cases. This may not be the optimal bootstrap formulation for reducing the bias of these parameters.

One other issue is the development of the grid for the Domingos' Error Decomposition problem, which is optimized for a two-dimensional problem, would need to be adjusted for a higher dimensional problem. One solution to this issue is to collect a fraction of the grid points but in higher dimensionality, which would maintain the computational cost of the problem.

Suggestions for Future Research

For future research, this methodology should be expanded to other sets of data and more complex algorithms. Decisions would need to be made for which order feature extraction and selection steps should be performed and which measures should be utilized. Any type of HSI data could be used for the subject of evaluation for the hierarchy. The individual utility value curves could be coded in a statistical software language to quickly assess the overall hierarchy values for each image. This can be compared to the use of the medians for each different factor level combination to see if these results remain the same.

One of the main advantages to the VFT methodology is the fact that it is flexible and modular for different problems and different decision makers. New decision makers should be surveyed to understand how the methodology would change for their inputs and the measures would be updated accordingly. The three basic branches for analysis could remain the same as they test the three most impactful measurements for the quality of the algorithms. More testing and validation should be accomplished at each increase of problem complexity to ensure that the weights used are still applicable.

Conclusions

This research was used as a way to fuse the quality assessment of many different images together using a value-focused hierarchy to determine the best algorithm to use in a certain situation. At times, a fusion of various different algorithms may provide better performance than a single classifier. This type of work has been accomplished before, and this value hierarchy can be modified to be used with different algorithms fused together. Additionally, the complexity of the hierarchy can be increased and decreased with the advice of the analyst and the decision makers in the process. Much like in previous research, different perspectives could change the

values and measures that are utilized within the hierarchy. The fusion of these different opinions could further strengthen the value of the output of these hierarchies. Additionally, different types of loss function decompositions could be utilized and analyzed for each different type of problem. Most of the measurements in this research were notional and chosen due to the uniqueness and newness of the type of analysis. In the real world, uniqueness may not be as critical of an objective and each analysis team must get together after careful deliberation to decide upon the values and measures of interest. Although CART was seen as the optimum algorithm for these test sets, it may not be the algorithm of choice for more complex problems.

Appendix A. Value Functions for Measures

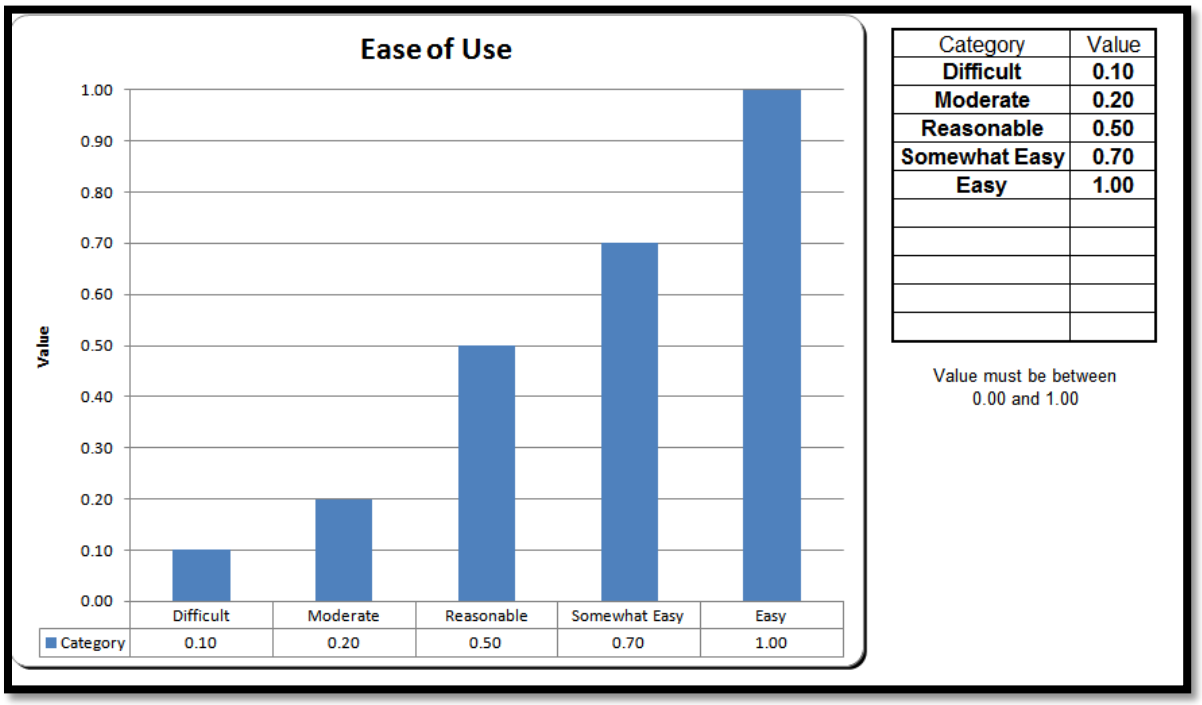


Figure 80. - Ease of Use Value Function

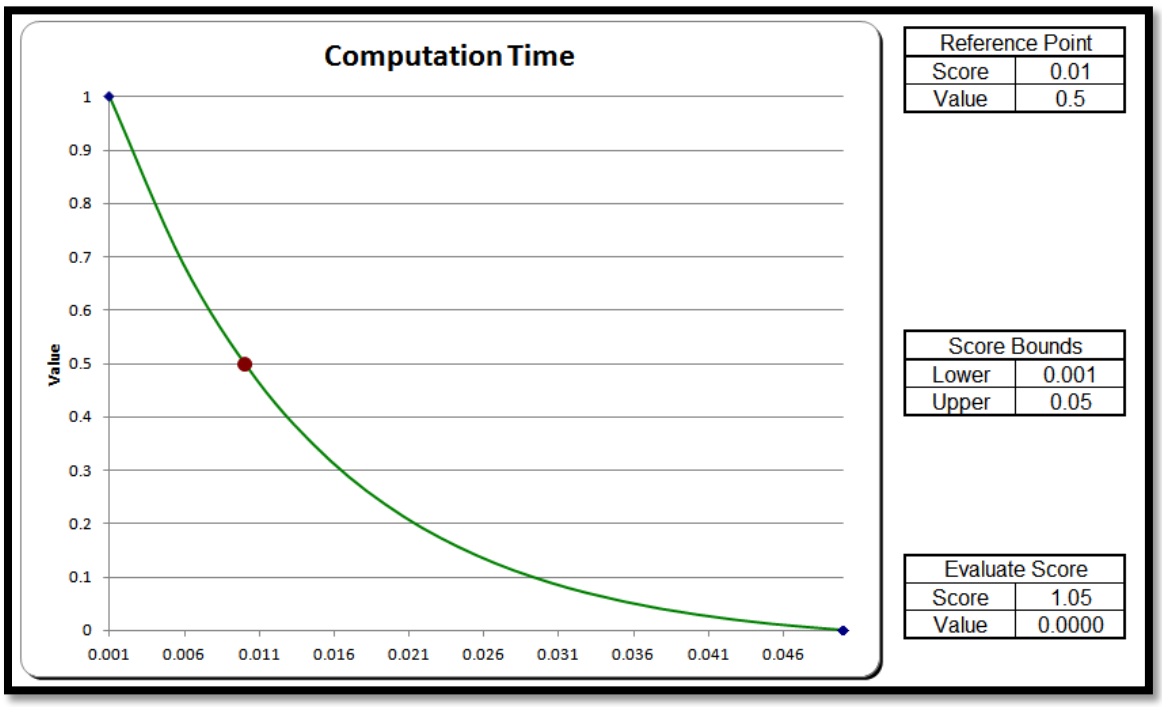


Figure 81. - Computation Time Value Function

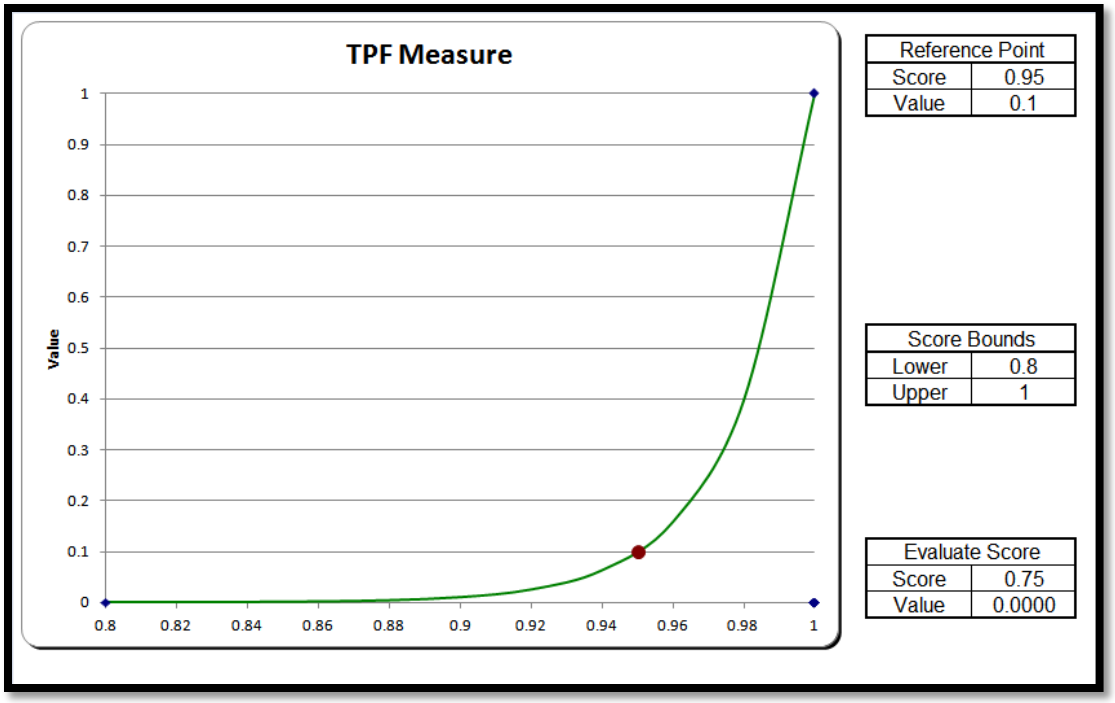


Figure 82. - TPF Measure Value Function

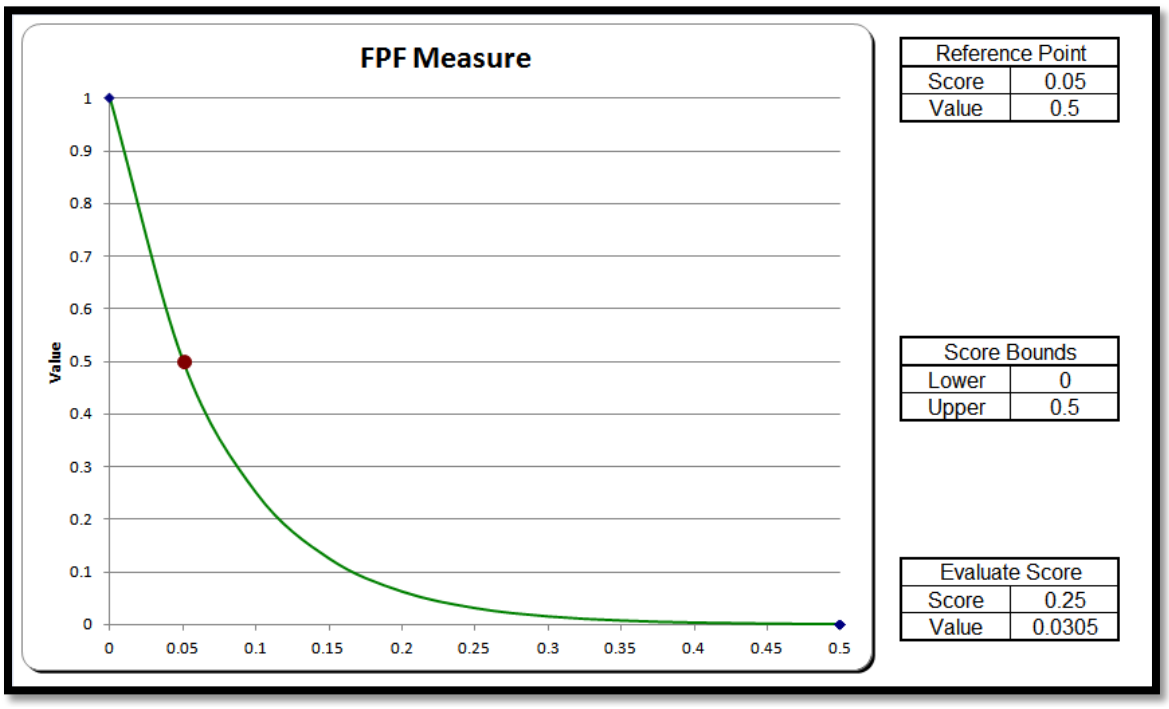


Figure 83. - FPF Measure Value Function

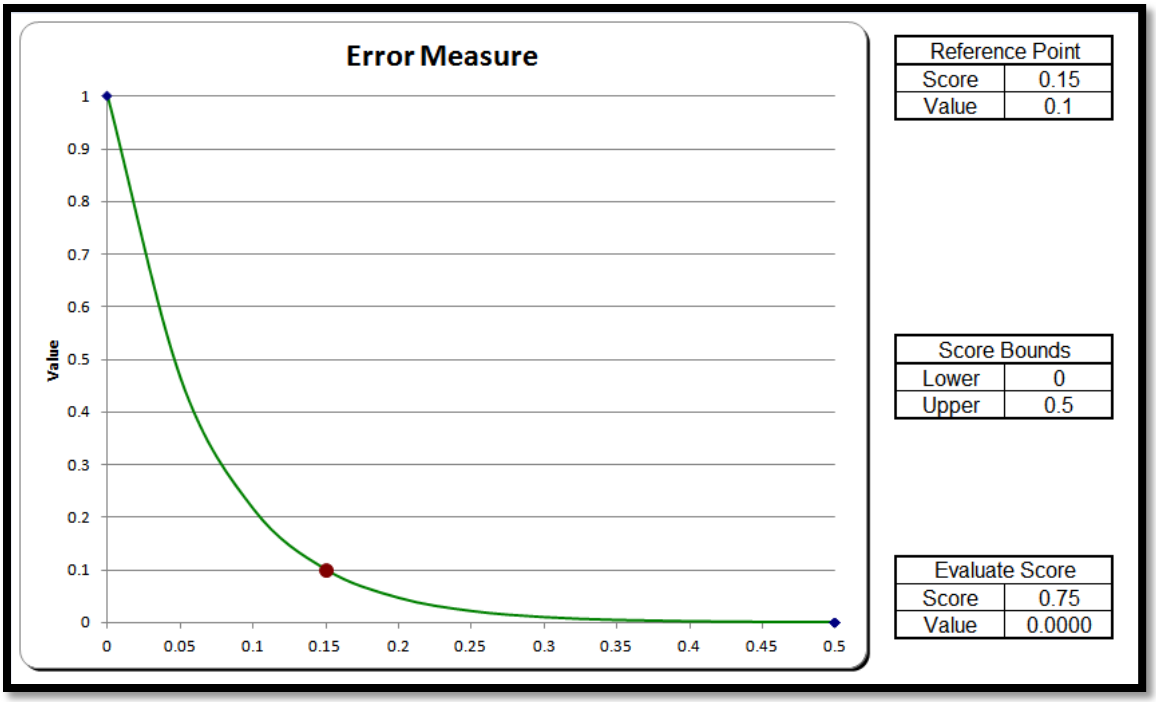


Figure 84. - Error Measure Value Function

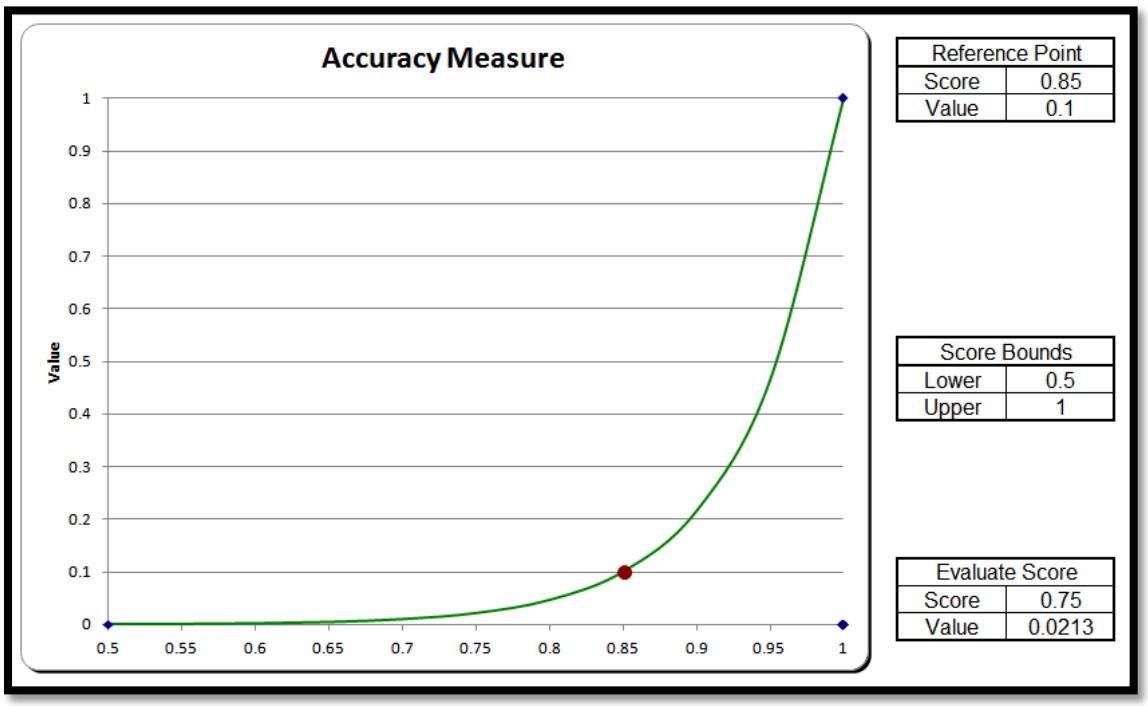


Figure 85. - Accuracy Measure Value Function

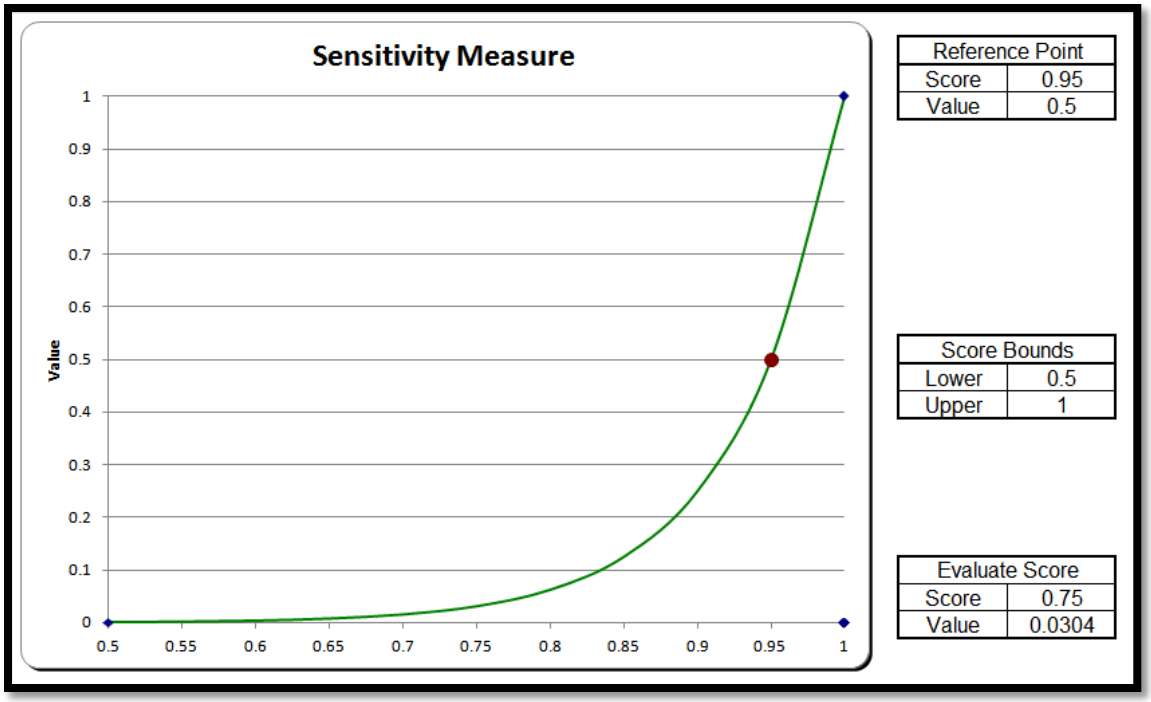


Figure 86. - Sensitivity Measure Value Function

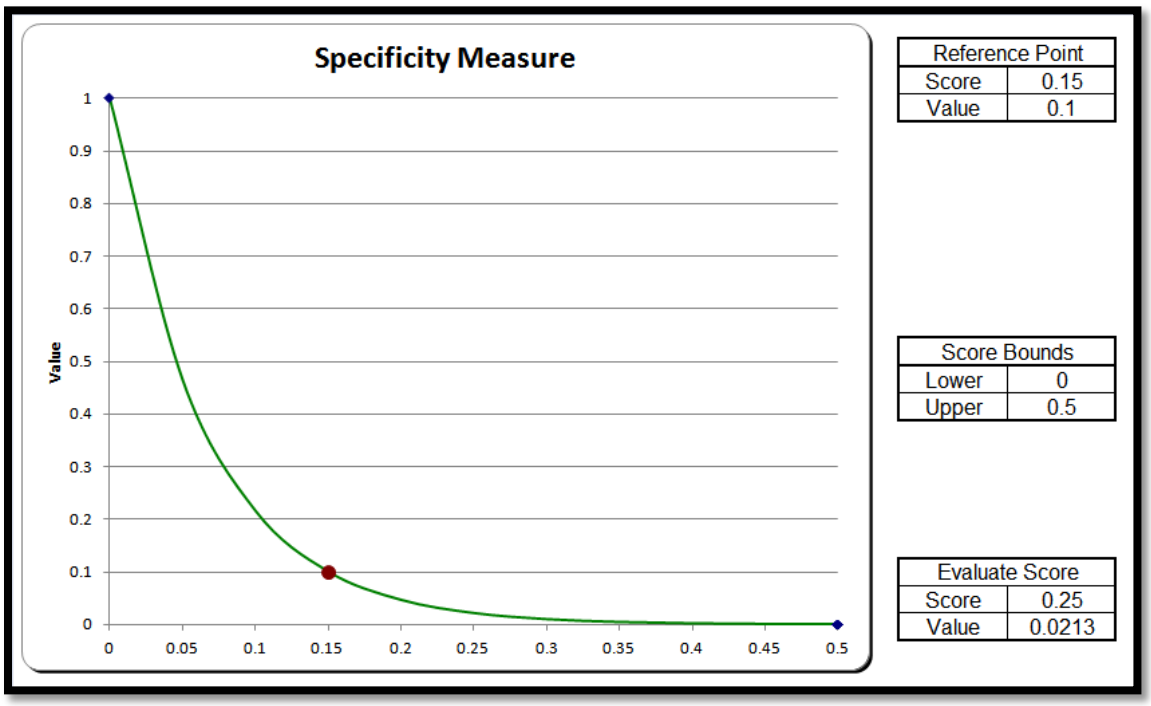


Figure 87. - Specificity Measure Value Function

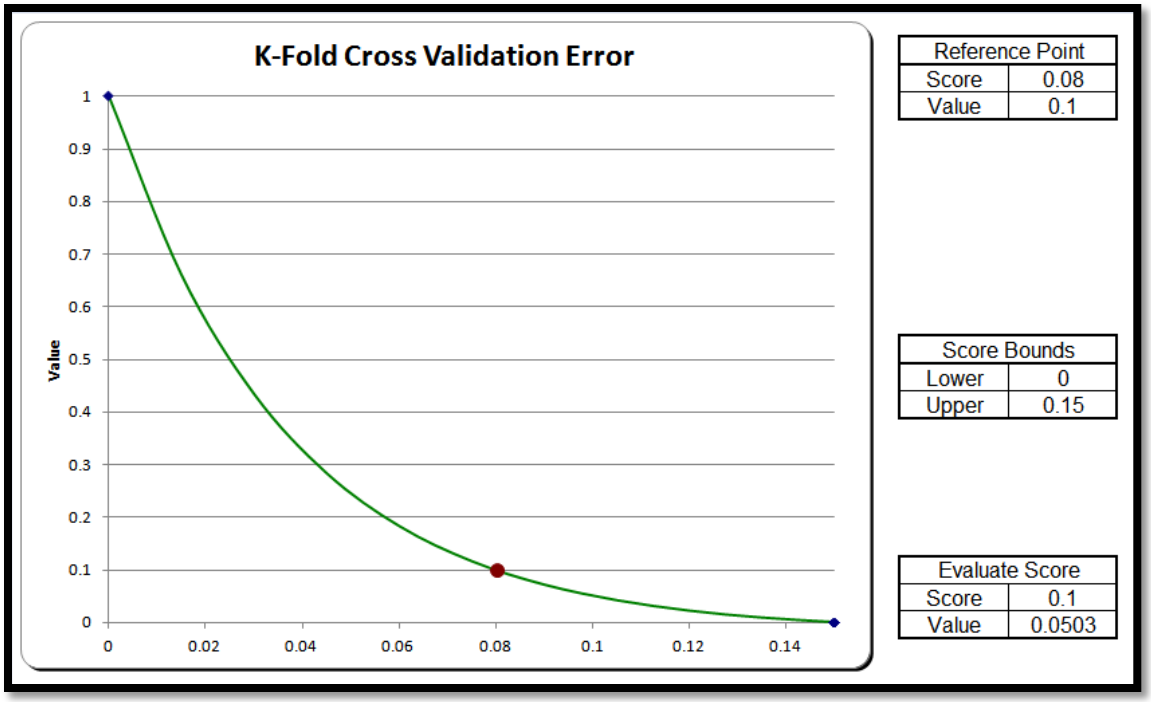


Figure 88. - K-fold Cross Validation Error Value Function

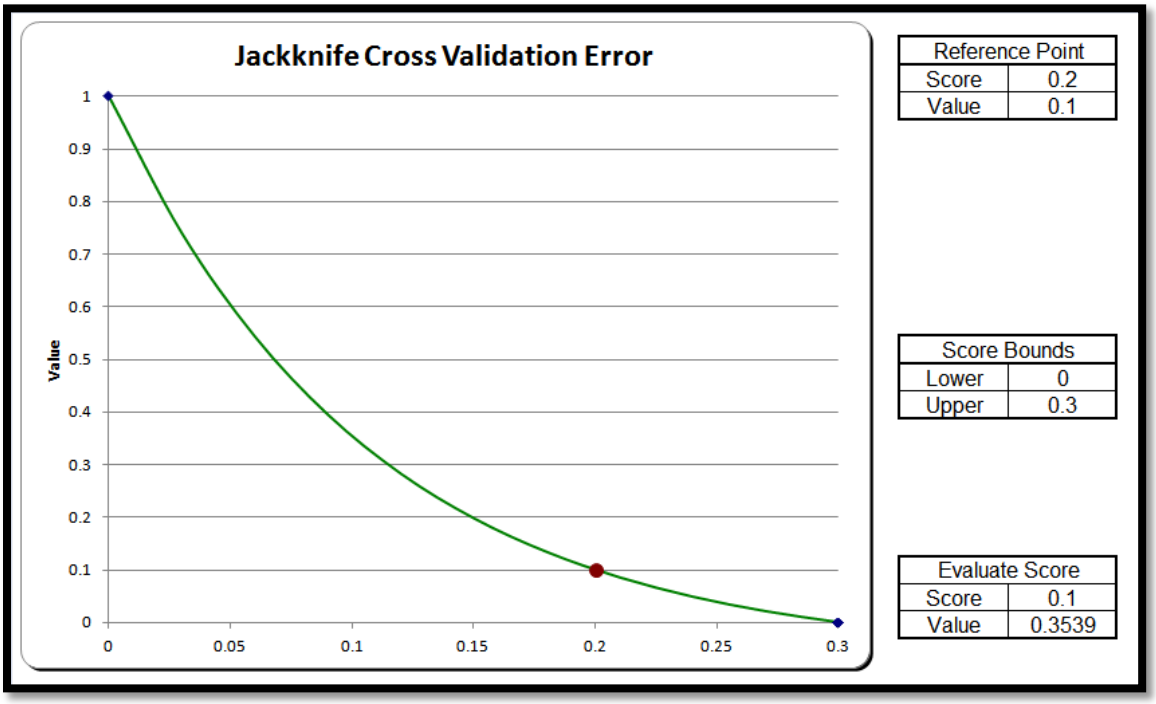


Figure 89. - Jackknife Cross Validation Error Value Function

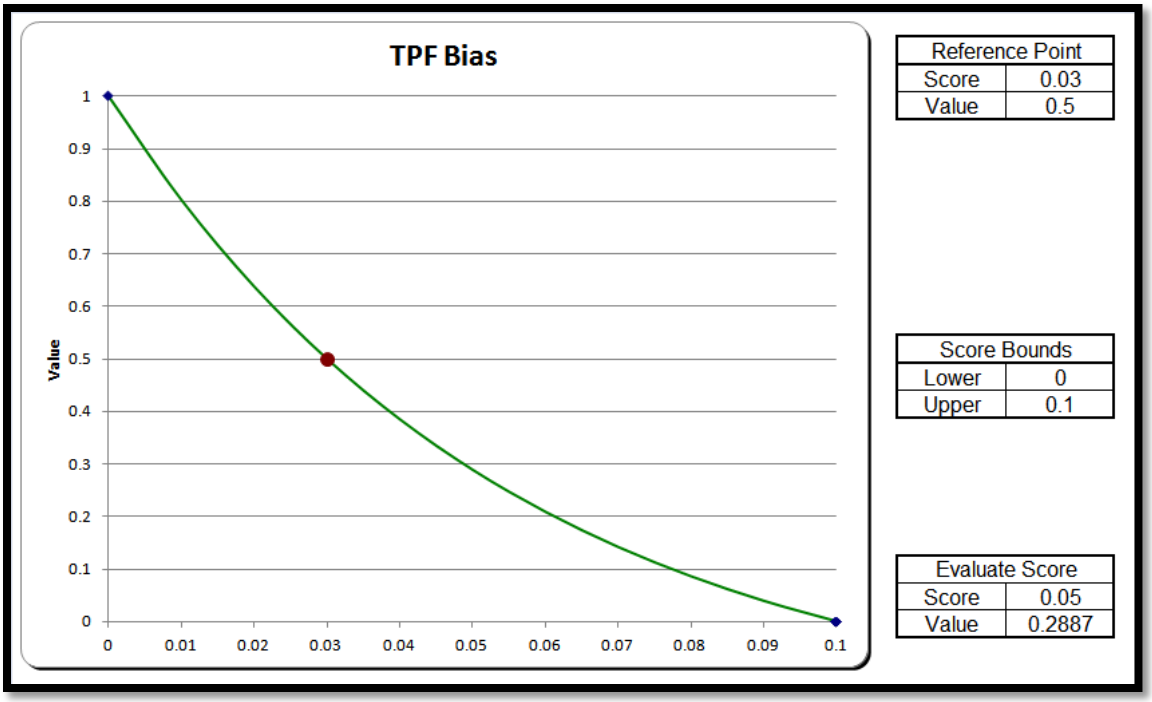


Figure 90. - TPF Bias Value Function

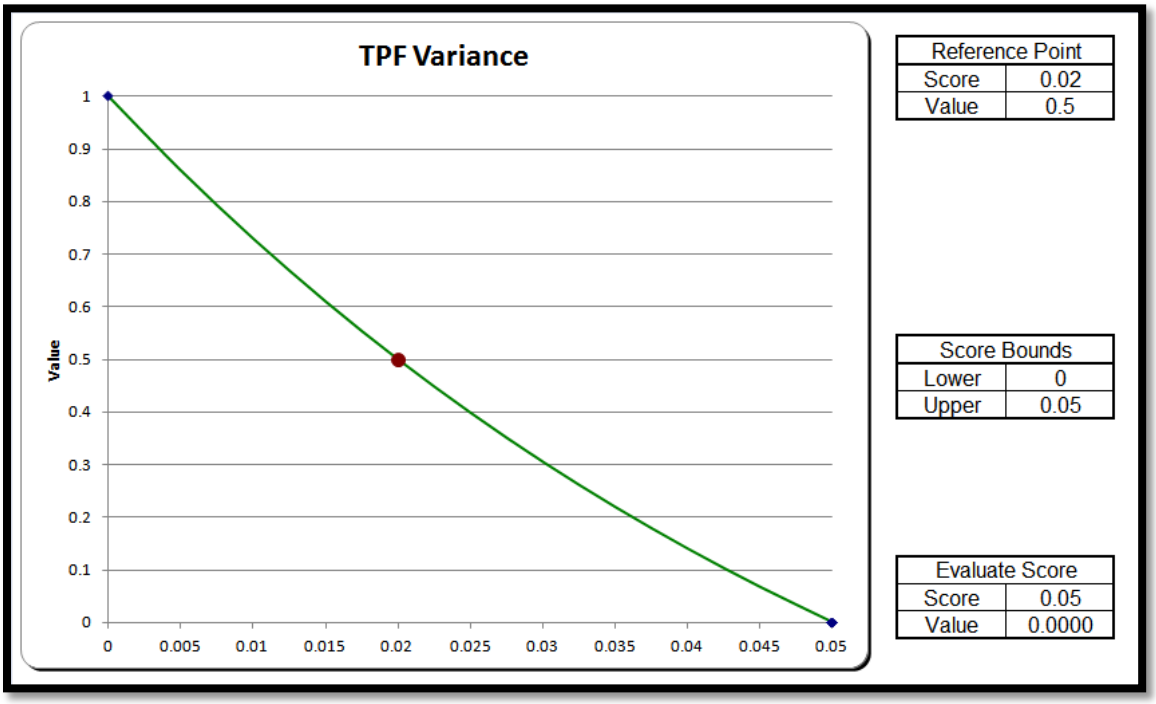


Figure 91. - TPF Variance Value Function

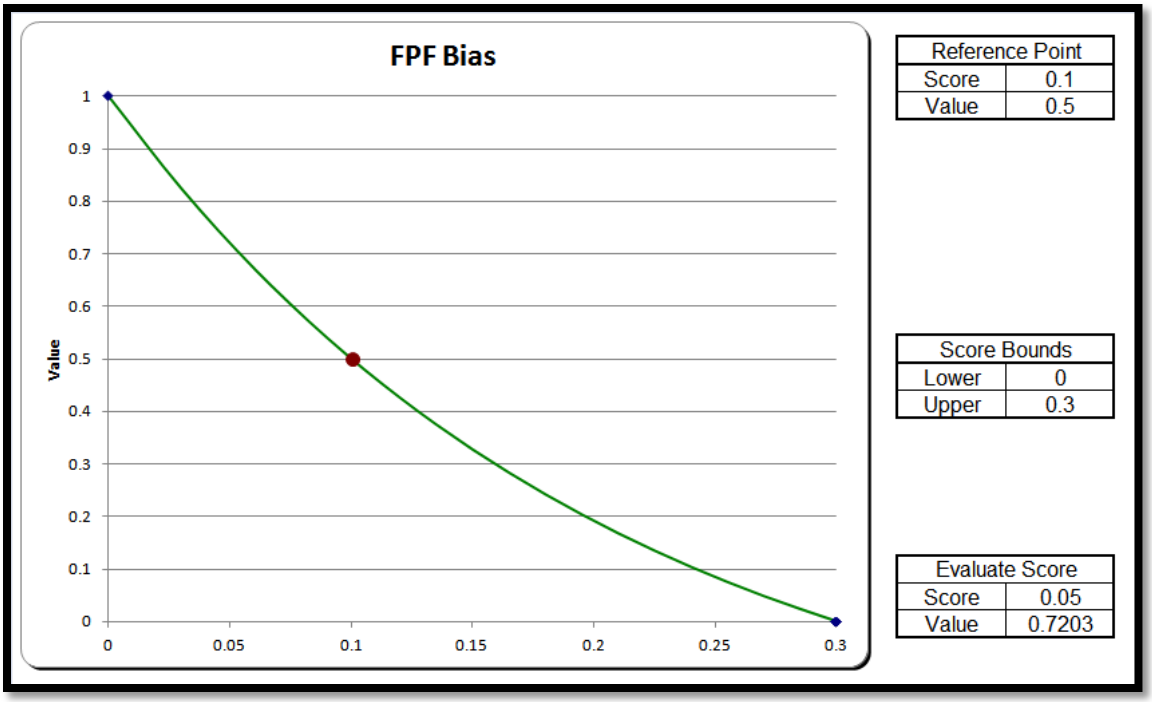


Figure 92. - FPF Bias Value Function

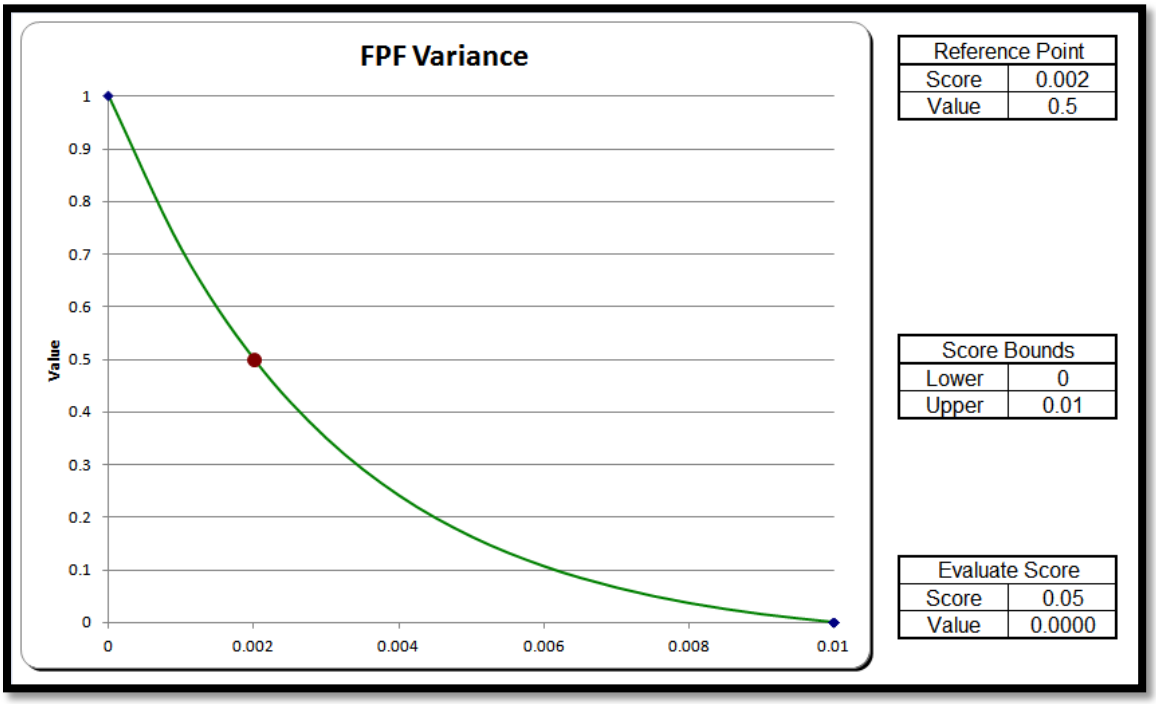


Figure 93. - FPF Variance Value Function

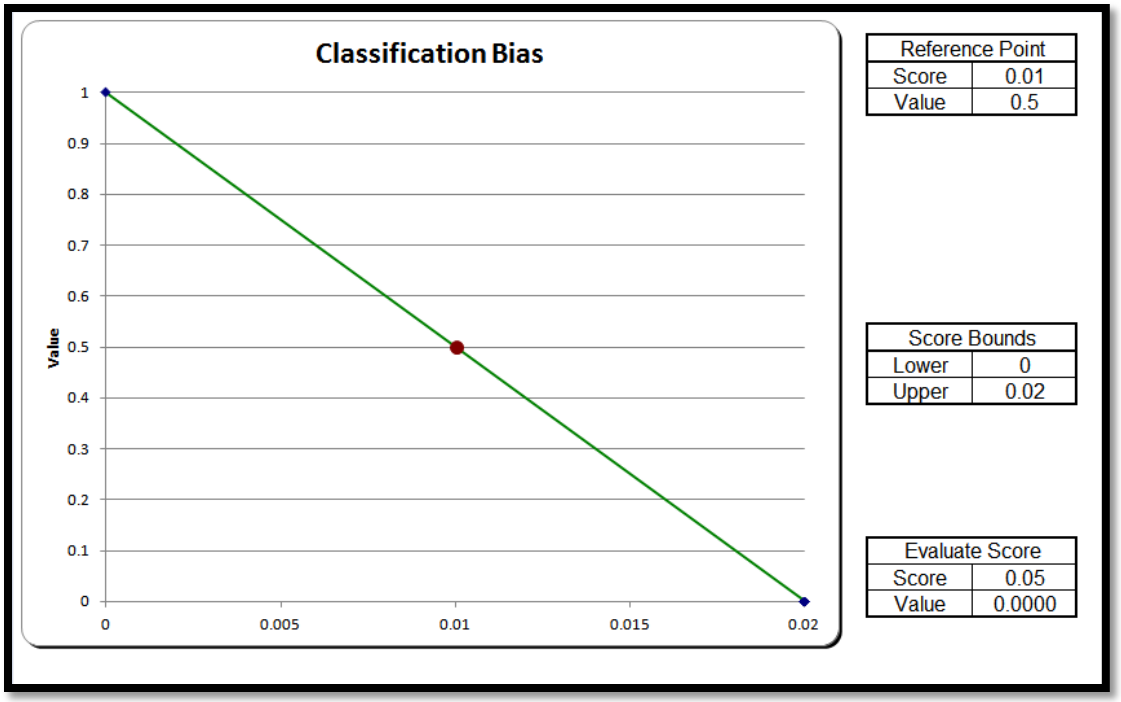


Figure 94. - Classification Bias Value Function

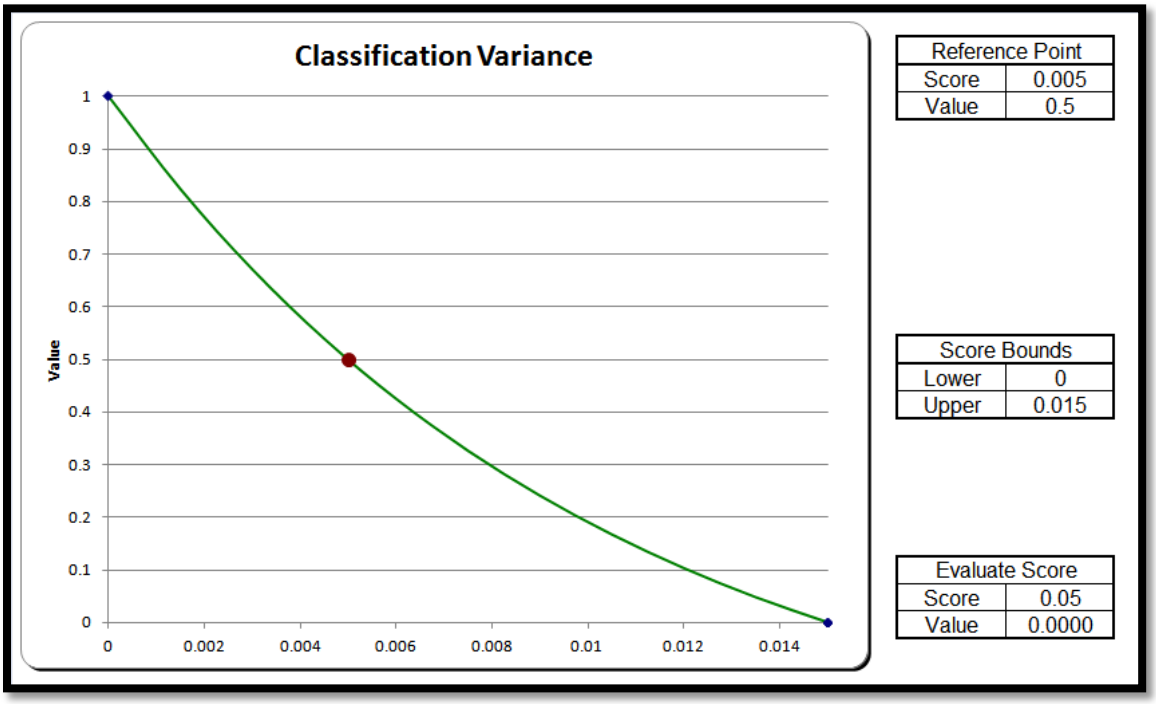


Figure 95. - Classification Variance Value Function

Appendix B. Bias and Variance Comparisons

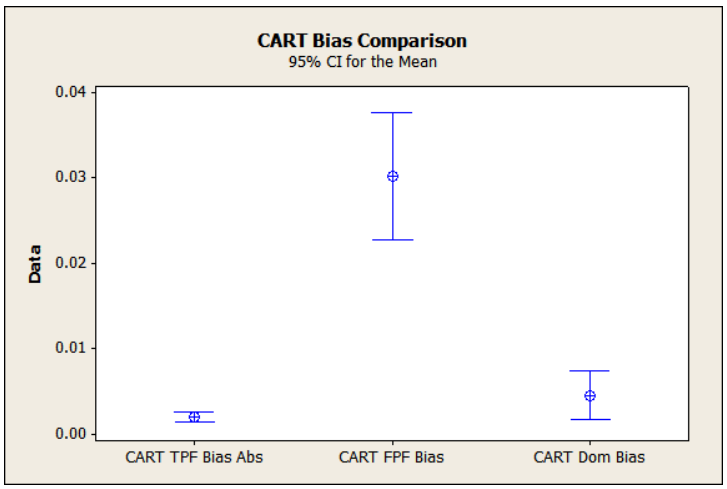


Figure 96. - CART Bias Comparison

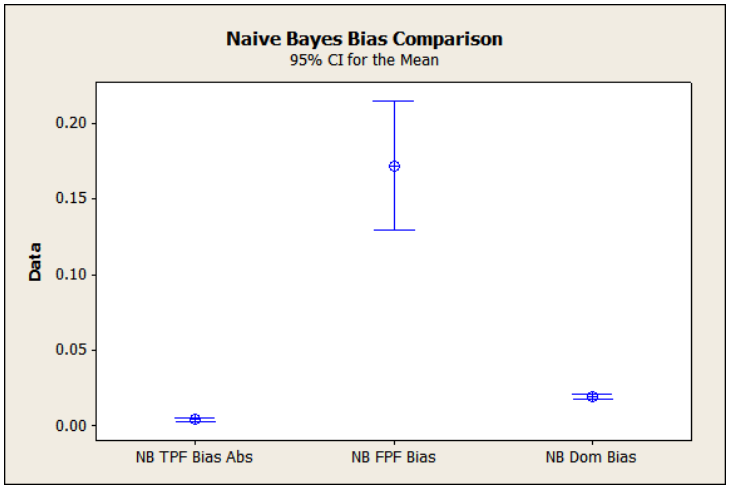


Figure 97. - Naive Bayes Bias Comparison

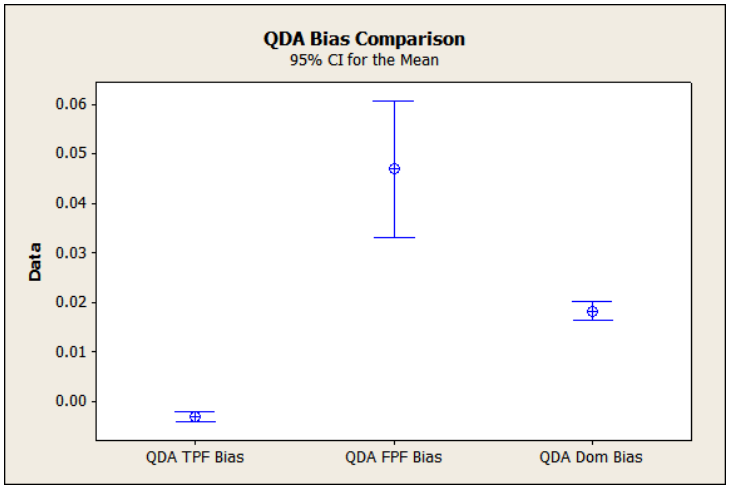


Figure 98. - QDA Bias Comparison

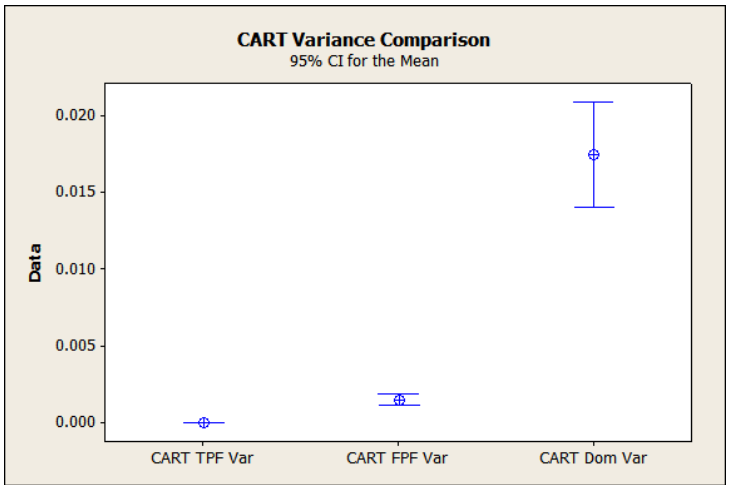


Figure 99. - CART Variance Comparison

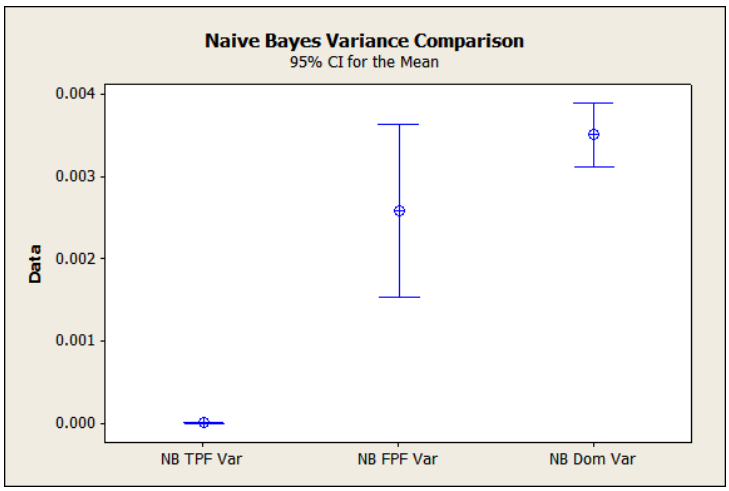


Figure 100. - Naive Bayes Variance Comparison

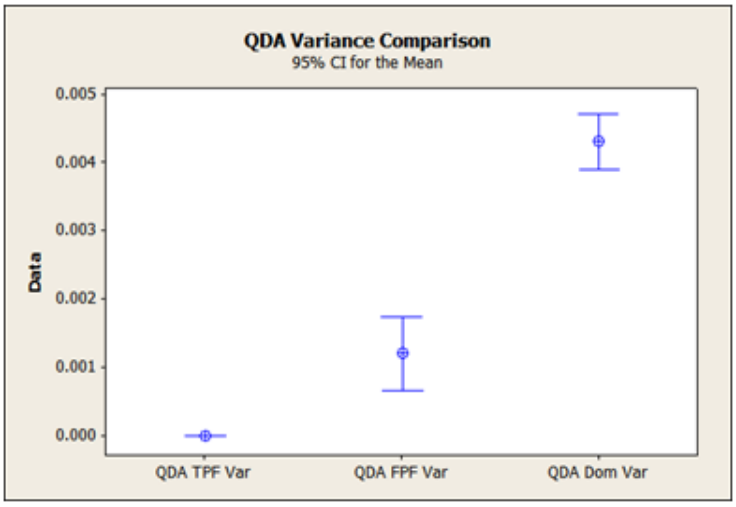


Figure 101. - QDA Variance Comparison

Appendix C. MATLAB Code

Main Data Compiler

```
%Initialize a data matrix for distances (y) and factors (x)
ParaMatrix=zeros(Comb,7);
DataMatrix=zeros(Comb*3,1600);
NBP=zeros(Comb,1);
BmuCell{Q}=cell(Comb,1);
TmuCell{Q}=cell(Comb,1);
BCMCell{Q}=cell(Comb,1);
TCMCell{Q}=cell(Comb,1);

rng('default')
%Automatically generates both MVG distributions and mean to distribution
%Mana. dists for each possible factor combination (need to adjust for
%different cov mtx and mus)
for PQi=1:3
    for TCi=1:3
        for BCi=1:3
            for Ti=1:3
                for Bi=1:3

%Create initial feature Matrix (Ones of pix qty/tgt qty n for two cols)
Ftrs=ones(PQm(PQi,2)+PQm(PQi,3),2);

%Set the Background or Target class response - number - Bgnd/Tgt Pix Qty
%This is based on setting the class to background for 1 to Bgnd Qty and the
%rest to target class
BorT=cell(PQm(PQi,2)+PQm(PQi,3),1);
for i=1:(PQm(PQi,2)+PQm(PQi,3));
    if i<PQm(PQi,3)+1;
BorT{i,1}='Background';
    else
BorT{i,1}='Target';
    end
end

%This is the same as above but for 0 and 1 (numerical) classes for bgnd and
%tgt
BorT2=zeros(PQm(PQi,2)+PQm(PQi,3),1);
for i=1:(PQm(PQi,2)+PQm(PQi,3));
    if i<PQm(PQi,3)+1;
BorT2(i,1)=0;
    else
BorT2(i,1)=1;
    end
end

%Develop the distributions for bgrnd/tgt
    BD = mvnrnd(Bmu(Bi,:),BCM(((2*BCi-1):2*BCi),:), PQm(PQi,3));
    TD = mvnrnd(Tmu(Ti,:),TCM(((2*TCi-1):2*TCi),:), PQm(PQi,2));
%Calculate the mahalanobis distance from mean to distribution
    TDBMmd= mahal(Bmu(Bi,:),TD);
    BDTMmd= mahal(Tmu(Ti,:),BD);
```

```
%Set the feature matrix to the distributions
    Ftrs(1:PQm(PQi,3),1:2)=BD;
    Ftrs(PQm(PQi,3)+1:end,1:2)=TD;

%Plug Distances, mean numbers, and cov mtx numbers into data matrix
ParaMatrix(Q,1)=TDBMmd;
ParaMatrix(Q,2)=BDTMmd;
ParaMatrix(Q,3)=Bi;
ParaMatrix(Q,4)=Ti;
ParaMatrix(Q,5)=BCi;
ParaMatrix(Q,6)=TCi;
ParaMatrix(Q,7)=PQm(PQi,1);
NBP(Q)=NP-NP*PQm(PQi,1);

%Create a matrix to hold all background/target distributions
DataMatrix(3*Q-2:3*Q-1,1:PQm(PQi,3))= BD';
DataMatrix(3*Q-2:3*Q-1,PQm(PQi,3)+1:NP)= TD';
DataMatrix(3*Q,:)= BorT2';

BmuCell{Q}=Bmu;
TmuCell{Q}=Tmu;
BCMCell{Q}=BCM;
TCMCell{Q}=TCM;
Q=Q+1;


                end
            end
        end
    end
end
```

Domingos' Bias/Variance Calculations

```
%Sets up all of the blank cells
CMTXx=cell(243,1);
CMTXy=cell(243,1);
Bmu=cell(243,1);
Tmu=cell(243,1);
BCM=cell(243,1);
TCM=cell(243,1);
CARTrepData=cell(243,50);
xGrid=cell(243,1);
xGridData=zeros(1000,2);
CARTyData=cell(243,50);
CARTyhatData=cell(243,50);
CARTCounts=cell(243,50);
tData=cell(243,1000);

%Develops the grid for testing for Domingos' Bias and Variance
    for i=1:Comb
    CMTXx{i}=DataMatrix(3*i-2:3*i-1,:)';
    CMTXy{i}=DataMatrix(3*i,:)';
    Bmu{i}=BmuCell{i};
    Tmu{i}=TmuCell{i};
```

```
    BCM{i}=BCMCell{i};
    TCM{i}=TCMCell{i};
        for j=1:1000;

xGridData(j,:)=[min(CMTXx{i}(:,1))+(j/1000)*range(CMTXx{i}(:,1)),(max(CMTXx{i
}(:,2))-min(CMTXx{i}(:,2))).*rand()+min(CMTXx{i}(:,2))];
        end
    xGrid{i}=xGridData;
    end

    %Develops The parametric bootstrapping for the unique run
    part=1;
    for i=1:Comb
        parfor j=1:50

        CARTrepData{i,j} =
CARTFuncD(CMTXx{i},CMTXy{i},Bmu{i},BCM{i},Tmu{i},TCM{i});
        end
    end

function [CARTrepData] = CARTFuncD(CMTXx,CMTXy,Bmu,BCM,Tmu,TCM)

  CARTmodel = fitctree(CMTXx,CMTXy);

   % Prune to a certain k, if there is not enough levels, don't prune
    if max(CARTmodel.PruneList)>5
        CARTmodelP= prune(CARTmodel,'level',5);
    else
        CARTmodelP=CARTmodel;
    end

  BDr = mvnrnd(Bmu,BCM,1200);
  TDr = mvnrnd(Tmu,TCM,400);
  test=[BDr;TDr];
  CARTyfitRep=predict(CARTmodelP,test);
  CARTrepData=[test,CARTyfitRep];

end

% Sets up the T and Y measurements
    part=2;
    for i=1:Comb
        parfor j=1:50

        [CARTtData{i,j},CARTyhatData{i,j},sizeT] =
CARTFuncY(CARTrepData{i,j}(:,1:2),CARTrepData{i,j}(:,3),xGrid{i})
        sizeT

        end
    end

function [CARTtData,CARThatData, sizeT] = CARTFuncY(CmtxX,CmtxY,xgrid)

CMTXc=[CmtxX(:,1:2),CmtxY];
```

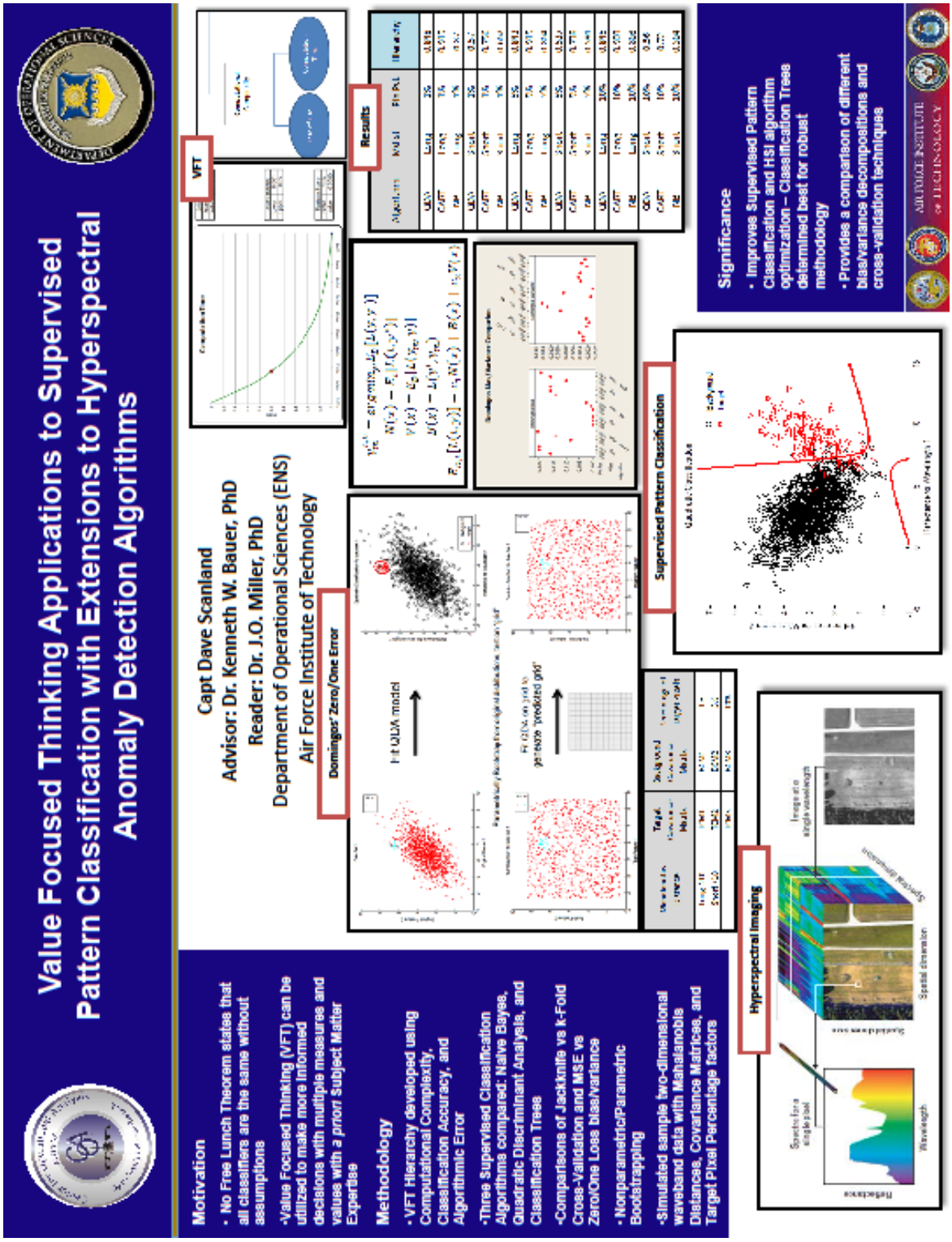
```
BMtx=CMTXc(find(CMTXc(:,3)==0),1:2);

TMtx=CMTXc(find(CMTXc(:,3)==1),1:2);
sizeT=size(TMtx,1)

if sizeT<2 || (rank(cov(BMtx))~=2 || rank(cov(TMtx))~=2)
  CARTtData=[];
  CARThatData=[];

else
    CARTmodel = fitctree(CMTXc(:,1:2),CMTXc(:,3));

   % Prune to a certain k, if there is not enough levels, don't prune
    if max(CARTmodel.PruneList)>5
        CARTmodelP= prune(CARTmodel,'level',5);
    else
        CARTmodelP=CARTmodel;
    end

  CARTtfitRep=predict(CARTmodelP,xgrid);
  CARTtData=[xgrid,CARTtfitRep];

end

if sizeT<2 || (rank(cov(BMtx))~=2 || rank(cov(TMtx))~=2)
  CARThatData=[];

    else
BMtx2=CARTtData(find(CARTtData(:,3)==0),1:2);

TMtx2=CARTtData(find(CARTtData(:,3)==1),1:2);
sizeT2=size(TMtx2,1)

    if sizeT2<2 || (rank(cov(BMtx2))~=2 || rank(cov(TMtx2))~=2)
  CARThatData=[];

    else
    CARTmodel2 = fitctree(xgrid,CARTtfitRep);

   % Prune to a certain k, if there is not enough levels, don't prune
    if max(CARTmodel.PruneList)>5
        CARThatmodel= prune(CARTmodel2,'level',5);
    else
        CARThatmodel=CARTmodel2;
    end

    CARThatfit=predict(CARThatmodel,xgrid);
    CARThatData=[xgrid,CARThatfit];
    end
end

end

    %Error checking for Domingos and calculation of Bias and Variance
```

```
        [EmptySet, CARTtData2, EmptySet2, CARTyhatData2] =
MakeEmpty(CARTtData,CARTyhatData);

function [EmptySet, QDAtData2, EmptySet2, QDAyhatData2] =
MakeEmpty(QDAtData,QDAyhatData);

%Sets empty cells
QDAtData2=QDAtData;
EmptyCell=cellfun('isempty',QDAtData2);
EmptyCell=EmptyCell(:,1);
Empties=zeros(243,2);
Empties(:,1)=EmptyCell;
Empties(:,2)=1:243;

QDAyhatData2=QDAyhatData;
EmptyCell2=cellfun('isempty',QDAyhatData2);
EmptyCell2sum=sum(EmptyCell2,2);

Empties2=zeros(243,2);
Empties2(:,1)=EmptyCell2sum;
Empties2(:,2)=1:243;

EmptyCellTest=EmptyCell2sum+EmptyCell;

%Finds the empty cells and removes them
EmptySet=Empties(find(Empties(:,1)==1),2)';

QDAtData2=QDAtData(find(EmptyCellTest(:,1)==0),:);

EmptySet2=Empties2(find(Empties2(:,1)>0),2)';

QDAyhatData2=QDAyhatData2(find(EmptyCellTest(:,1)==0),:);

end

[Stackedt, yhatStacked, tsize] = Stacked(CARTtData2,CARTyhatData2);

function [Stackedt, yhatStacked, tsize] = Stacked(QDAtData2,QDAyhatData2);

%Sets up new matrices
tsize=size(QDAtData2,1);
tNewData=cell(tsize,50);
Stackedt=cell(tsize,1);

yhatNewData=cell(tsize,50);
yhatStacked=cell(tsize,1);

%Stacks up the t matrices and sets into Stackedt
for i=1:tsize
    for j=1:50
 tNewData{i,j}=QDAtData2{i,j}';
 Stackedt{i}(j,:)=tNewData{i,j}(3,:);
    end
end
```

```
%Does the same for the y matrices
for i2=1:tsize
    ii=i2
    for j2=1:50

    yhatNewData{i2,j2}=QDAyhatData2{i2,j2}';
    yhatStacked{i2}(j2,:)=yhatNewData{i2,j2}(3,:);
    jj=j2
    end
end

end

        [DomData, DomAve] = DomingosBV(Stackedt, yhatStacked, tsize);

function [DomData, DomAve] = DomingosBV(Stackedt, yhatStacked, tsize);

%Finds the Mode of the t which is the optimum prediction
Modet=cell(tsize,1);
for i=1:tsize
    for j=1:1000
        Modet{i,1}(1,j)=mode((Stackedt{i,1}(:,j)));
        ii=i
        jj=j
    end
end

%Finds the mode of the y which is the main prediction
Modey=cell(tsize,1);
for i2=1:tsize
    for j2=1:1000
        Modey{i2,1}(1,j2)=mode((yhatStacked{i2,1}(:,j2)));
        ii2=i2
        jj2=j2
    end
end

%Sets up the Bias, Variance, and Noise cells
BiasD=cell(tsize,1);
VarD=cell(tsize,1);
NoiseD=cell(tsize,1);

%Carries out the Domingos' calculations
for i3=1:tsize
    for j3=1:1000

        BiasD{i3,1}(1,j3)=abs(Modet{i3,1}(1,j3)-Modey{i3,1}(1,j3));

        VarD{i3,1}(1,j3)=mean(abs(Modey{i3,1}(1,j3)-
yhatStacked{i3,1}(:,j3)));

        NoiseD{i3,1}(1,j3)=mean(abs(Stackedt{i3,1}(:,j3)-Modet{i3,1}(1,j3)));
```

```
    end
end

%Calculates the Probopt to decide the c1 and c2 variables and then
%calculates the total expected loss
qnum=cell(tsize,1);
Probopt=cell(tsize,1);
c1=cell(tsize,1);
c2=cell(tsize,1);
ExpLoss=cell(tsize,1);
for i4=1:tsize
    for j4=1:1000
           qnum{i4,1}(1,j4)=sum(yhatStacked{i4,1}(:,j4)==Modet{i4,1}(1,j4));
           Probopt{i4,1}(1,j4)=qnum{i4,1}(1,j4)/50;
           c1{i4,1}(1,j4)=2*Probopt{i4,1}(1,j4)-1;
           if Modey{i4,1}(1,j4)==Modet{i4,1}(1,j4);
               c2{i4,1}(1,j4)=1;
           else
               c2{i4,1}(1,j4)=-1;
           end

ExpLoss{i4,1}(1,j4)=c1{i4,1}(1,j4)*NoiseD{i4,1}(1,j4)+BiasD{i4,1}(1,j4)...
                              +c2{i4,1}(1,j4)*VarD{i4,1}(1,j4);

    end

end
```

Appendix D. Quad Chart



Bibliography

Bassham, C. B. (2002). *Automatic Target Recognition Classification System Evaluation Methodology* (No. AFIT/DS/ENS/02-03). AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT.

Bassham, B., Bauer, K. W., & Miller, J. O. (2006). Automatic Target Recognition System Evaluation Using Decision Analysis Techniques. *Military Operations Research*, *11*(1), 49-66.

Borghys, D., Kåsen, I., Achard, V., & Perneel, C. (2012). Comparative Evaluation of Hyperspectral Anomaly Detectors in Different Types of Background. *SPIE Defense, Security, and Sensing* (pp. 83902J-83902J). International Society for Optics and Photonics.

Camps-Valls, G., & Bruzzone, L. (Eds.). (2009). *Kernel Methods for Remote Sensing Data Analysis* (Vol. 26). New York: Wiley.

Domingos, P. (2000). A Unified Bias-Variance Decomposition. *Proceedings of 17th International Conference on Machine Learning. Stanford CA* (pp. 231-238).

Domingos, P. (2000). A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. *AAAI/IAAI*, *2000*, 564-569.

Domingos, P. (2012). A Few Useful Things to Know About Machine Learning. *Communications of the ACM*, *55*(10), 78-87.

Dougherty, G. (2013). *Pattern Recognition and Classification: An Introduction*. Springer Science & Business Media.

Dube. W. (2009). *Remote Sensing.* SPIE Professional.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons.

Eismann, M. T. (2012). *Hyperspectral Remote Sensing*. Bellingham: SPIE Press.

Fortmann-Roe, S. (2014). Understanding the Bias-Variance Tradeoff. http://scott.fortmann-roe.com/docs/BiasVariance.html. Accessed Feb 2015.

Friedman, J. H. (1997). On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality. *Data Mining and Knowledge Discovery*, *1*(1), 55-77.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., & Tibshirani, R. (2009). *The Elements of Statistical Learning* (Vol. 2, No. 1). New York: Springer.

Johnson, R. A., & Wichern, D. W. (2014). *Applied Multivariate Statistical Analysis*. Pearson Education Limited.

Keeney, R. L. (1996). Value-Focused Thinking: Identifying Decision Opportunities and Creating Alternatives. *European Journal of Operational Research*, *92*(3), 537-549.

Keeney, R. L. (2009). *Value-Focused Thinking: A Path to Creative Decisionmaking*. Harvard University Press.

Kirkwood, C. W. (1996). *Strategic Decision Making*. Wadsworth Publ. Co..

Dietterich, T. G., & Kong, E. B. (1995). *Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithms*. Technical Report, Department of Computer Science, Oregon State University.

Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons.

Manolakis, D. G., Shaw, G. A., & Keshava, N. (2000, August). Comparative Analysis of Hyperspectral Adaptive Matched Filter Detectors. *AeroSense 2000* (pp. 2-17). International Society for Optics and Photonics.

Manolakis, D. G., Marden, D., Kerekes, J. P., & Shaw, G. A. (2001, August). Statistics of Hyperspectral Imaging Data. *Aerospace/Defense Sensing, Simulation, and Controls* (pp. 308-316). International Society for Optics and Photonics.

Manolakis, D., & Shaw, G. (2002). Detection Algorithms for Hyperspectral Imaging Applications. *Signal Processing Magazine, IEEE*, *19*(1), 29-43.

Manolakis, D., Marden, D., & Shaw, G. A. (2003). Hyperspectral Image Processing for Automatic Target Detection Applications. *Lincoln Laboratory Journal*, *14*(1), 79-116.

Manolakis, D., Lockwood, R., Cooley, T., & Jacobson, J. (2009, May). Is There a Best Hyperspectral Detection Algorithm?. In *SPIE Defense, Security, and Sensing* (pp. 733402-733402). International Society for Optics and Photonics.

McGee, C. M. (2003). *A Value Focused Thinking Approach to Software Interface in a Complex Analytical Domain* (No. AFIT/GOR/ENS/03-16). AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT.

Orloff, S. & B. Weinberg, G. Shaw, S.M. Hsu, C. Upham, J. Evans, K. Heinemann (2000). *Investigation of Supervised Background Classification Algorithms on Hyperspectral Data.* Project Report HTAP-4, Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, May 2000.

Paciencia, T. J. (2014). *Improving Non-Linear Approaches to Anomaly Detection, Class Separation, and Visualization* (Doctoral dissertation, AIR FORCE INSTITUTE OF TECHNOLOGY).

Pirsig, R. M. (1999). *Zen and the Art of Motorcycle Maintenance: An Inquiry Into Values*. Random House

Raschka, S. (2014). Naive Bayes and Text Classification I-Introduction and Theory. *arXiv preprint arXiv:1410.5329*.

Raschka. S. (2015). Seminar. Practical Data Science: An Introduction to Supervised Machine Learning and Pattern Classification: The Big Picture. Conference: ICER NextGen Bioinformatics Seminar Series 2015, At Michigan State University https://speakerdeck.com/rasbt/practical-data-science-an-introduction-to-supervised-machine-learning-and-pattern-classification-the-big-picture Accessed Feb 2015.

Rickard, L.J. & R. Basedow, P. P. Silverglate and E. E. Zalewski (1993). *HYDICE: An Airborne System for Hyperspectral Imaging. Proc. of the SPIE,* vol. 1937, pp. 173-179.

Shalizi, C. (2010). The Bootstrap. *American Scientist*, *98*(3), 186-190.

Shalizi, C. (2011). *The Bootstrap. Advanced Data Analysis.* http://www.stat.cmu.edu/~cshalizi/402/lectures/08-bootstrap/lecture-08.pdf. Accessed Feb 2015.

Sharma, D., Yadav, U. B., & Sharma, P. (2009). The Concept of Sensitivity and Specificity in Relation to Two Types of Errors and its Application in Medical Research. *Journal of Reliability and Statistical Studies (ISSN: 0974-8024)*, *2*(2), 53-58.

Shaw, G., & Manolakis, D. (2002). Signal Processing for Hyperspectral Image Exploitation. *Signal Processing Magazine, IEEE*, *19*(1), 12-16.

Shoviak, M. J. (2001). *Decision Analysis Methodology to Evaluate Integrated Solid Waste Management Alternatives for a Remote Alaskan Air Station* (No. AFIT/GEE/ENV/01M-20). AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH.

Shaw, G. A., & Burke, H. H. K. (2003). Spectral Imaging for Remote Sensing. *Lincoln Laboratory Journal*, *14*(1), 3-28.

Tomaselli, V., Guarnera, M., Marchisio, C. D., & Moro, S. (2013, March). Low Complexity Smile Detection Technique for Mobile Devices. In *IS&T/SPIE Electronic Imaging* (pp. 86610O-86610O). International Society for Optics and Photonics.

Weir, J. D. (2015). Hierarchy_Builder.xlsm. Retrieved Feb 2015. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT.

Wolpert, D. H., & Macready, W. G. (1995). *No Free Lunch Theorems for Search* (Vol. 10). Technical Report SFI-TR-95-02-010, Santa Fe Institute.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
26-03-2015	Master's Thesis	Oct 2013 – March 2015

TITLE AND SUBTITLE

Value Focused Thinking Applications to Supervised Pattern Classification With Extensions to Hyperspectral Anomaly Detection Algorithms

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Scanland, David S., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENS)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-MS-15-M-121

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally left blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Statement A.
Approved for Public Release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Hyperspectral imaging (HSI) is an emerging analytical tool with flexible applications in different target detection and classification environments, including Military Intelligence, environmental conservation, etc. Algorithms are being developed at a rapid rate, solving various related detection problems under certain assumptions. At the core of these algorithms is the concept of supervised pattern classification, which trains an algorithm to data with enough generalizability that it can be applied to multiple instances of data. It is necessary to develop a logical methodology that can weigh responses and provide an output value that can help determine an optimum algorithm. This research focuses on the comparison of supervised learning classification algorithms through the development of a value focused thinking (VFT) hierarchy. This hierarchy represents a fusion of qualitative/ quantitative parameter values developed with Subject Matter Expert a priori information. Parameters include a fusion of bias/variance values decomposed from quadratic and zero/one loss functions, and a comparison of cross-validation methodologies and resulting error. This methodology is utilized to compare the aforementioned classifiers as applied to hyperspectral imaging data. Conclusions reached include a proof of concept of the credibility and applicability of the value focused thinking process to determine an optimal algorithm in various conditions.

15. SUBJECT TERMS

Value Focused Thinking (VFT); Supervised Pattern Classification; Hyperspectral Imaging (HIS); Domingos' Zero/One Loss Bias/Variance Decomposition; Cross-Validation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Kenneth W. Bauer (ENS)
U	U	U	UU	177	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636, ext 4328 Kenneth.Bauer@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18