

Air Force Institute of Technology AFIT Scholar

Theses and Dissertations

Student Graduate Works

9-15-2011

Distributed Spacing Stochastic Feature Selection and its Application to Textile Classification

Jeffrey D. Clark

Follow this and additional works at: <https://scholar.afit.edu/etd>

Part of the [Atomic, Molecular and Optical Physics Commons](#), and the [Other Materials Science and Engineering Commons](#)

Recommended Citation

Clark, Jeffrey D., "Distributed Spacing Stochastic Feature Selection and its Application to Textile Classification" (2011). *Theses and Dissertations*. 1370.
<https://scholar.afit.edu/etd/1370>

This Dissertation is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.



DISTRIBUTED SPACING STOCHASTIC FEATURE SELECTION AND ITS
APPLICATION TO TEXTILE CLASSIFICATION

DISSERTATION

Jeffrey D. Clark, Lt. Col., USAF

AFIT/DEE/ENG/11-05

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT/DEE/ENG/11-05

DISTRIBUTED SPACING STOCHASTIC FEATURE SELECTION AND ITS
APPLICATION TO TEXTILE CLASSIFICATION

DISSERTATION

Presented to the Faculty
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 Doctor of Philosophy

Jeffrey D. Clark, B.S.E., M.S.E.E.

Lt. Col., USAF

September 2011

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/DEE/ENG/11-05

DISTRIBUTED SPACING STOCHASTIC FEATURE SELECTION AND ITS
APPLICATION TO TEXTILE CLASSIFICATION

Jeffrey D. Clark, B.S.E., M.S.E.E.
Lt. Col., USAF

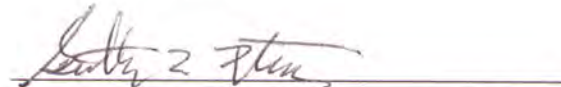
Approved:



Major Michael J. Mendenhall, PhD
(Chairman)

31-AUG-2011

Date



Dr. Gilbert L. Peterson (Member)

31 AUG 2011

Date

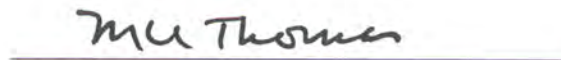


Dr. Matthew C. Fickus (Member)

31 Aug 2011

Date

Accepted:



M. U. Thomas
Dean, Graduate School of Engineering
and Management

9 Sep 2011

Date

Abstract

Many situations require the need to quickly and accurately locate dismounted individuals in a variety of environments. In conjunction with other dismount detection techniques, being able to detect and classify clothing (textiles) provides a more comprehensive and complete dismount characterization capability. Because textile classification depends on distinguishing between different material types, hyperspectral data, which consists of several hundred spectral channels sampled from a continuous electromagnetic spectrum, is used as a data source. However, a hyperspectral image generates vast amounts of information and can be computationally intractable to analyze. A primary means to reduce the computational complexity is to use feature selection to identify a reduced set of features that effectively represents a specific class. While many feature selection methods exist, applying them to continuous data results in closely clustered feature sets that offer redundancy and fail in the presence of noise. This dissertation presents a novel feature selection method that limits feature redundancy and improves classification. This method uses a stochastic search algorithm in conjunction with a heuristic that combines measures of distance and dependence to select features. Comparison testing between the presented feature selection method and existing methods uses hyperspectral data and image wavelet decompositions. The presented method produces feature sets with an average correlation of 0.40-0.54. This is significantly lower than the 0.70-0.99 of the existing feature selection methods. In terms of classification accuracy, the feature sets produced outperform those of other methods, to a significance of 0.025, and show greater robustness under noise representative of a hyperspectral imaging system.

Acknowledgements

I would like to thank Dr. Michael Mendenhall and Dr. Gilbert Peterson and Dr. Matthew Fickus of the Air Force Institute of Technology for their help and support of this research. I would also like to thank my family for their support and encouragement throughout this process. I especially want to thank my wife for her continuing love, understanding, and support (FAANMW).

Jeffrey D. Clark

Table of Contents

	Page
Abstract	iv
Acknowledgements	v
List of Figures	viii
List of Tables	xv
List of Algorithms	xvii
List of Symbols	xviii
I. Introduction	1-1
1.1 Background	1-4
1.2 Characteristics of Fabrics	1-4
1.3 Hyperspectral Data	1-6
1.4 Feature Selection	1-8
1.5 The Problem	1-10
1.6 Methodology	1-10
1.7 Organization	1-10
II. Background	2-1
2.1 Feature Selection	2-1
2.2 Current Methods for Feature Selection	2-4
2.2.1 Other Specific Related Works	2-6
2.3 Classifier and Feature Selection Method Qualifications	2-8
2.4 Target Detection	2-8
2.5 Search Algorithms and Statistics	2-14
2.6 Redundancy in Feature Sets	2-20
2.7 Summary	2-22
III. Feature Selection and Detection	3-1
3.1 Non-correlated Aided Simulated Annealing Feature Selection Overview	3-2
3.1.1 NASAFS Methodology	3-4
3.1.2 Final Feature Selection Process	3-22
3.2 Non-correlated Aided Simulated Annealing Feature Selection - Integrated Distribution Function Overview	3-27

	Page
3.2.1	NASAFS-IDF Methodology 3-28
3.2.2	Final Feature Selection Process 3-34
3.3	Correlation Detection Method 3-37
3.4	Summary 3-42
IV.	Experimental Results and Analysis 4-1
4.1	Comparison Tests 4-1
4.1.1	Data 4-3
4.1.2	Configurations of Test 4-5
4.2	NASAFS & NASAFS-IDF Parameters 4-8
4.3	Results of Experiment 4-11
4.3.1	Textile Data Results 4-11
4.3.2	LCVF Data Results 4-20
4.3.3	Texture Data Results 4-25
4.4	Significance of Feature Selection Methods 4-33
4.5	Summary 4-36
V.	Conclusion 5-1
5.1	Summary of Results 5-2
5.2	Recommendations for Future Work 5-3
Appendix A.	Wavelet Decomposition A-1
Bibliography 5

List of Figures

Figure		Page
1.1.	Featured clockwise from top left: (1) Search and Rescue, (2) surveillance, (3) anti-camouflage capability, (3) target identification and tracking. These are examples of different target detection scenarios where clothing detection enhances mission objectives [2,78].	1-2
1.2.	Examples of knitting (left) and weaving (right) that lead to different spectral signatures [88,89].	1-5
1.3.	Hyperspectral data is most easily understood as an image; two dimensions define the spatial location and the third dimension is a vector of spectral values. Above is a hyperspectral image showing the different spectral information contained in hyperspectral data [65].	1-7
1.4.	Above is an example of a textile sample, where each sample is a different color of the same material: black (dashed), brown (dotted), and green (solid). The plot exemplifies how similar the spectral characteristics are at longer wavelengths, even if the different colors make the sample dramatically different in the visible spectrum.	1-8
2.1.	Schematic of the classifiers, based on their neural net categorization, as determined by Holmstrom <i>et al.</i> [35], and added to by González [38].	2-9
2.2.	Example of a generic Receiver Operating Characteristic (ROC) curve.	2-11
2.3.	Flow chart for the simulated annealing search process.	2-18
2.4.	Example of the redundancy produced by the Bhattacharyya feature set selection method. The diamonds are the feature locations of the six features chosen by the Bhattacharyya method, shown on the textile signal 80% Polyester 20% Rayon.	2-21

Figure		Page
3.1.	Example of a hyperspectral signal segment, divided into bins and sub-regions. The solid line is the reference signal and the dashed line is the test signal; 1...3 are the sub-regions and A...I are the bins. Both signals have a $1nm$ sampling interval and each bin is constructed of 10 spectral attributes.	3-5
3.2.	Diagram depicting the flow of feature selection method NASAFS (within the dashed box), and its connection to a classification method (outside the dashed box).	3-6
3.3.	Example of a hyperspectral signal segment; divided into bins and sub-regions. The low-resolution data uses a sliding window technique instead of a sequential binning process. Each letter corresponds to a different bin; the solid line is the reference signal, and the dashed line is the test signal, and A...E are the bins. The example shown is for a $10nm$ bin size consisting of 10 spectral attributes per bin.	3-7
3.4.	Correlation matrix of the 12 class textile data set. The lines denote the different sub-regions and the average correlation value of that region is shown in the box. The row and column variables for the sub-region locator method are 50 and 100, respectively.	3-15
3.5.	This figure illustrates the placement of the markers for determining the sub-regions of the correlation matrix. The row and column variables for the sub-region locator method are 50 and 100, respectively.	3-16
3.6.	Correlation matrix of the 12 class textile data set. The lines denote the different sub-regions, where the row and column variables for the sub-region locator method are 10 and 25, respectively.	3-17
3.7.	Flow chart for the heuristic function of NASAFS.	3-19

Figure	Page	
3.8.	Histogram of selected features for: (a) 65% Polyester / 35% Cotton vs 80% Nylon / 20% Spandex, (b) 65% Polyester / 35% Cotton vs 94% Polyester / 6% Spandex, (c) 65% Polyester / 35% Cotton vs 100% Cotton, and (d) 65% Polyester / 35% Cotton vs 100% Silk. The red lines are the divisions of the sub-regions. The green line is the 0.05 (A_{val}) criteria line. The sub-regions labeled A - E have an average correlation coefficient of 0.9823, 0.6688, 0.7066, 0.4348, and 0.3936 respectively. The feature set selection process is indicated by the labeled circles for a feature set size of seven.	3-26
3.9.	Diagram depicting the flow of feature selection method NASAFS-IDF (within dashed line), and its connection to a classification method (outside dashed line).	3-29
3.10.	This is an example of the final feature selection process for 100% Cotton Woven, with acceptable distributed spacing set to a 35% optimal distribution. The order of feature assignment is indicated; green boxes indicate the attributes that are occluded due to feature selection.	3-37
3.11.	Histogram created by NASAFS-IDF, and used by final feature selection stage, to determine a feature set for the reference class. The dashed line indicates feature selection. This histogram is of the first class of the texture data set.	3-38
3.12.	The top and bottom number lines indicate the domain of a sample with 100 dimensions. The numbers above each number line indicate the location of the i^{th} feature. The top number line indicates placement of features for acceptable distribution, whereas the bottom number line indicates one possible placement for an 80% acceptable distribution. The distance of 16 indicates the minimum number of dimensions between features for this percent acceptable distribution. The feature set size for both cases is 6.	3-39
3.13.	This is an example of the averaging technique for the CoDeM process.	3-39

Figure		Page
3.14.	Flow chart for the detection algorithm Correlation Detection Method (CoDeM).	3-40
4.1.	Representative samples from the 12 class textile data set: 65% Polyester 35% Cotton Woven (red), 80% Nylon 20% Spandex Knit (green), 97% Bamboo 3% Spandex Knit (tan), 100% Cotton Woven (blue), 100% Polyester Woven (black), 100% Satin Woven (pink), and 100% Silk Woven (brown).	4-4
4.2.	Select representative spectra of the Lunar Crater Volcanic Field (LCVF) 23 class data set, classes A (red), G (green), H (orange), L (magenta), O (purple), Q (black), and R (blue). The water absorption bands are indicated by the vertical dotted lines [61,62].	4-5
4.3.	Brodatz samples with their associated class labels [87].	4-6
4.4.	Diagram of two hyperspectral signals. The top figure depicts a clean hyperspectral signal of 65% Polyester 35% Cotton blend; the bottom figure depicts the hyperspectral signal for 65% Polyester 35% Cotton blend, with additive white Gaussian noise of a 0.03 power level.	4-7
4.5.	Pictorial diagram showing the process used to obtain the correlation coefficient value for each feature set.	4-8
4.6.	Illustration of the chance selects, based on the initial temperature of the simulated annealing process. Diagrams show results (left) using an initial temperature of 0.02, (mid) using an initial temperature of 0.4, and (right) using an initial temperature of 1.0.	4-11
4.7.	Reflectance spectra for 65% Polyester 35% Cotton blend (dashed) and 80% Nylon 20% Spandex blend (solid) signals with discriminating feature sets of four independent NASAFS runs: run 1 – square, run 2 – diamond, run 3 – asterisk, run 4 – circle.	4-13
4.8.	Accuracy of NASAFS, using CoDeM, of different sized feature sets (ranging from 4 to 36 features) over 4 noise realizations (from 0 to 0.03) for the 12 class textile data set.	4-14

Figure		Page
4.9.	Accuracy of NASAFS-IDF, using CoDeM, of different sized feature sets (ranging from 2 to 25 features) over 4 noise realizations (from 0 to 0.03) for the 12 class textile data set.	4-15
4.10.	Diagram comparing average correlation coefficient of the feature set versus the feature set size for the 12 class textile data set. This is at 30% acceptable distribution for NASAFS-IDF.	4-16
4.11.	Results of the classification accuracy for the 12 class textile data set using CoDeM. Each feature selection method is represented with a different color: NASAFS (red), NASAFS-IDF (magenta), ReliefF (blue), GRLVQI (green), and Bhattacharyya (black).	4-17
4.12.	Hyperspectral signal for 80% Polyester 20% Rayon blend. The respective feature sets chosen by each feature selection method are indicated: NASAFS (box), NASAFS-IDF (X), ReliefF (circle), GRLVQI (asterisk), and Bhattacharyya (diamond).	4-18
4.13.	Pareto Front for the 12 class textile data set using CoDeM for the 30% acceptable distributed spacing criteria. The Pareto Front is for accuracy versus feature set size and is indicated by circles connected by lines.	4-19
4.14.	Contingency table for 12 class textile data set as reported by Naïve Bayes, using the feature sets of NASAFS-IDF with a 30% acceptable distributed spacing. C.A. is the consumers accuracy, C.E. is the consumer error, P.A. is the producers accuracy and O.E. is the omission error.	4-20
4.15.	Contingency table for 12 class textile data set as reported by Naïve Bayes, using the feature set of class 7 of NASAFS-IDF with a 30% acceptable distributed spacing.	4-21
4.16.	Correlation matrix of the 23 class LCVF data set. Sub-regions are determined by NASAFS and are marked with solid vertical black lines. The correlation coefficient of each sub-region is shown in the boxes.	4-22
4.17.	Representative sample of the LCVF 23 class data set, with the locations of the features of the feature sets selected by NASAFS (diamond), NASAFS-IDF (star), ReliefF (square), GRLVQI (circle), and Bhattacharyya (dot).	4-23

Figure	Page
4.18. Pareto Front for the LCVF 23 class data set using CoDeM for the 45% acceptable distributed spacing criteria. The Pareto Front is accuracy versus feature set size and is indicated by circles connected by lines.	4-24
4.19. Contingency table for LCVF 23 class data set as reported by Naïve Bayes, using the feature sets of NASAFS-IDF with a 45% acceptable distributed spacing. C.A. is the consumers accuracy, C.E. is the consumer error, P.A. is the producers accuracy and O.E. is the omission error.	4-25
4.20. Contingency table for 23 class LCVF data set as reported by Naïve Bayes, using the feature set of class 1 of NASAFS-IDF with a 45% acceptable distributed spacing.	4-26
4.21. Correlation matrix of the 7 class texture data set.	4-27
4.22. Accuracy of NASAFS-IDF, using CoDeM, of different sized feature sets (ranging from 2 to 25 features) over 4 noise realizations (from 0 to 0.03) for the 7 class texture data set for a 35% acceptable distribution.	4-27
4.23. Diagram comparing the average correlation coefficient of the feature set versus the feature set size for the 7 class texture data set at 35% acceptable distribution for NASAFS-IDF.	4-28
4.24. Selected representative sample for the 7 class texture data set.	4-29
4.25. Class 1 and 2 sample of the 7 class texture data set. The features of each feature selection method are indicated as follows: NASAFS-IDF star, ReliefF diamond, GRLVQI circle, and Bhattacharyya 'x'.	4-30
4.26. Class 1 and 2 sample of the 7 class texture data set. The features of each feature selection method are indicated as follows: NASAFS-IDF star, ReliefF diamond, and GRLVQI circle. . . .	4-31
4.27. Pareto Front for the 7 class texture data set using CoDeM for the 35% acceptable distributed spacing criteria. The Pareto Front is accuracy versus feature set size and is indicated as circles connected by lines.	4-32

Figure		Page
4.28.	Contingency table for 7 class texture data set as reported by Naïve Bayes, using the feature sets of NASAFS-IDF with a 35% acceptable distributed spacing. C.A. is the consumers accuracy, C.E. is the consumer error, P.A. is the producers accuracy and O.E. is the omission error.	4-32
4.29.	Contingency table for 7 class texture data set as reported by Naïve Bayes, using the feature set of class 2 of NASAFS-IDF with a 35% acceptable distributed spacing.	4-33
A.1.	Example of a Morlet wavelet [91].	A-2
A.2.	Example of a signal processed by a generic wavelet transformation. The top portion is the high-pass filter; the bottom portion is the low-pass filter [63].	A-3
A.3.	Example of a wavelet transformation, where $h[n]$ represents the low-pass filter, and $g[n]$ represents the high-pass filter.	A-3
A.4.	Example of a wavelet decomposition, where $h[n]$ represents the low-pass filter, and $g[n]$ represents the high-pass filter.	A-3
A.5.	Example of a two-dimensional wavelet transform, where cA_n is the approximation coefficient for the n^{th} level, and $cD_n^{(\beta)}$ is the detail coefficient for the n^{th} level. β is either the horizontal (h), diagonal (d), or vertical (v) component [64].	A-4
A.6.	Daubechies wavelet of the 8^{th} order, where a.) is the wavelet function, b.) is the scaling function, c.) is the digital low-pass filter, and d.) is the digital high-pass filter [71].	A-4

List of Tables

Table		Page
4.1.	Class types of the 12 class textile data set.	4-3
4.2.	NASAFS and NASAFS-IDF feature selection parameters. . . .	4-12
4.3.	Average correlation coefficients of four single runs of NASAFS comparing 65% Polyester 35% Cotton blend to 80% Nylon 20% Spandex blend.	4-12
4.4.	Accuracy and Average Correlation Coefficients for NASAFS, NASAFS-IDF, ReliefF, GRLVQI, and Bhattacharyya methods for the 12 class textile data set, where the feature size is six with no noise added to the data. The bin size for NASAFS-IDF is $10nm$ and the acceptable distributed spacing set to 30%.	4-18
4.5.	Accuracy and Average Correlation Coefficients for NASAFS, NASAFS-IDF, ReliefF, GRLVQI, and Bhattacharyya methods for the LCVF data set, where the feature size is six with no noise added to the data. The bin size for NASAFS-IDF was $10nm$ and the acceptable distributed spacing is set to 45%.	4-23
4.6.	Accuracy and Average Correlation Coefficients for NASAFS-IDF, ReliefF, GRLVQI, and Bhattacharyya methods for the 7 class texture data set, where the feature size is six with no noise added to the data. The bin size for NASAFS-IDF was $10nm$ and the acceptable distributed spacing is set to 35%.	4-29
4.7.	Wilcoxon signed-rank test results for the feature selection methods tested in this work. The row is considered as method <i>A</i> and the column is considered as method <i>B</i> . For the Wilcoxon signed-rank test, and as our null hypothesis states for this test, we are determining if method <i>A</i> is better than method <i>B</i> with a significance better than 0.05.	4-36

Table		Page
4.8.	Wilcoxon signed-rank test results for all five feature selection methods tested in this work. The row is considered as method <i>A</i> and the column is considered as method <i>B</i> . For the Wilcoxon signed-rank test, and as our null hypothesis states for this test, we are determining if method <i>A</i> is better than method <i>B</i> with a significance better than 0.05.	4-36

List of Algorithms

1	Simulated Annealing pseudo-code [74]	3-23
2	Non-correlated Aided Simulated Annealing Feature Selection (NASAFS) pseudo-code	3-25
3	Non-correlated Aided Simulated Annealing Feature Selection - Inte- grated Distribution Function (NASAFS-IDF) pseudo-code	3-36

List of Symbols

Symbol		Page
N	Dimensionality of the data	2-1
H_0	Null hypothesis	2-9
H_1	Alternative hypothesis	2-9
P_D	Probability of detection	2-10
P_F	Probability of false alarm	2-10
T	Simulated annealing cooling temperature	2-17
σ^2	Variance	2-19
cov	Covariance	2-19
xcov	Cross-covariance	2-19
r	Correlation	2-20
R	Autocorrelation	2-20
r_X	Correlation matrix	3-4
M	Total number of samples	3-4
m	Total number of bins of a sample	3-4
F	Feature set	3-4
ι_u	Acceptable distribution	3-4
h	Heuristic for NASAIFS	3-4
ρ	Cross-correlation	3-6
k_h	Covariance threshold for heuristic	3-7
q	Number of sub-regions	3-10
$ F_{sub(i)} $	Number of selected feature points in the i^{th} sub-region	3-10
$ \overline{F}_{sub(i)} $	Expected number of features in the i^{th} sub-region	3-10
σ	Standard deviation	3-10
$n_{sub(i)}$	Number of dimensions of the i^{th} sub-region	3-11
$w_{sub(s)}$	Weighting of smaller sub-region	3-11

Symbol		Page
β	Bin size maximum limit	3-11
$ F $	Cardinality of feature set	3-12
ι	Distributed spacing	3-14
\bar{r}	Average correlation	3-18
R	Autocorrelation	3-20
D	Distance	3-20
d_h	Distance threshold for heuristic	3-20
H_{hit}	Number of times a feature is selected across all histograms	3-23
A_{val}	Number of times a feature is selected in a histogram	3-23
n_i	Number of dimensions between features for NASAFS-IDF	3-30
f_i	The i^{th} feature of the feature set	3-30
\tilde{n}	Number of dimensions expected between features	3-30
h_N	Heuristic for NASAFS-IDF	3-32
γ	Distributed spacing ratio for NASAFS-IDF	3-32
A	Weight applied to γ	3-32
k_{CoDeM}	Cross-covariance threshold for CoDeM	3-38
d_{CoDeM}	Distance threshold for CoDeM	3-38
T_{int}	Initial temperature	4-8
T_{final}	Final temperature	4-8
T_{decay}	Decay rate of T	4-8
N_{bin}	Bin size	4-8
NdB	Noise power	4-8
$APct$	Percent acceptable distribution	4-8

DISTRIBUTED SPACING STOCHASTIC FEATURE SELECTION AND ITS APPLICATION TO TEXTILE CLASSIFICATION

I. Introduction

The process used to locate something usually requires knowing certain aspects of that thing. For example, if told to locate a specific box in a room full of boxes, it is helpful to know the size, shape, or color of the target box. Limited knowledge of the characterization aspects of that box reduces the possibility of a successful outcome. More generally, the less information obtained about the target of detection, the greater the difficulty in completing the objective.

This work focuses on textile detection in relation to the identification and location of people. Detection of people, otherwise known as dismount detection, has many applications. Some examples of this include: Search and Rescue, surveillance, anti-camouflage, and target identification and tracking (Fig. 1.1). All of these areas have civilian and military applications that would be enhanced by improvements in passive identification of discriminating dismount characteristics.

Accurate detection of an object requires obtaining knowledge about that object. The same is true for dismount detection. There are many methods of dismount detection, and as many methods of avoiding it. Therefore, it is advantageous to assemble as much information about a dismount as possible, in order to increase opportunities for accurate detection. Collecting information about a dismount's physical characteristics (height, weight, and hair color, etc.) can be useful. Accurate passive identification requires the combination of multiple dismount characteristics, including textile identification.

To understand the usefulness of textile identification, consider the following examples. The first, involves the passive location of a militant leader, separating him from a large group of fellow combatants. While the basic physical characteristics

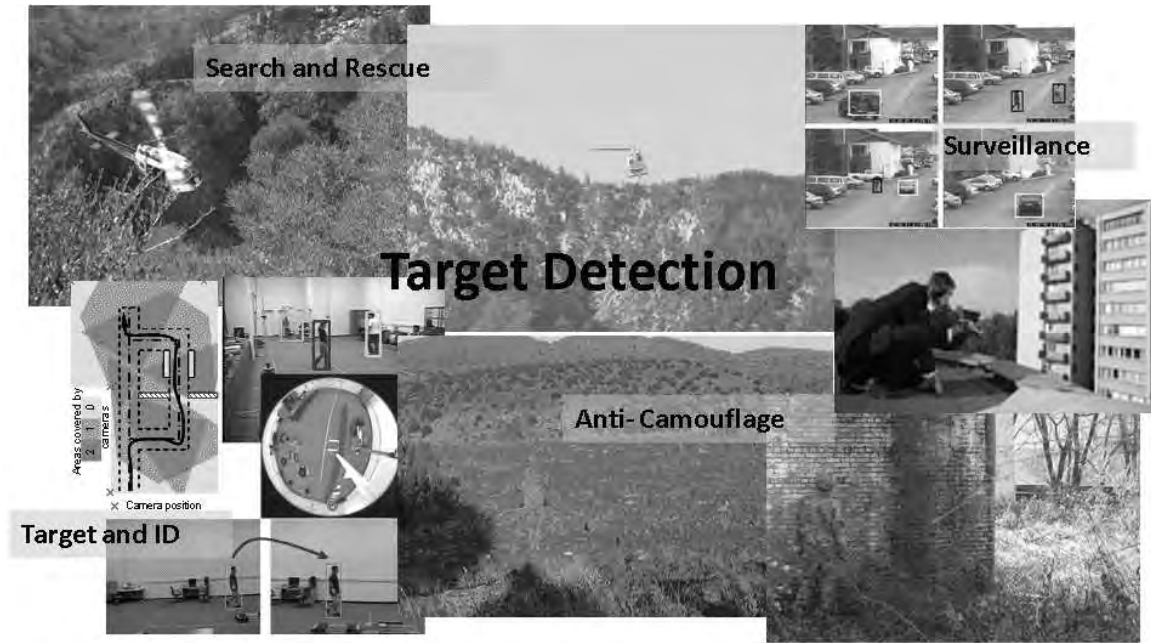


Figure 1.1: Featured clockwise from top left: (1) Search and Rescue, (2) surveillance, (3) anti-camouflage capability, (3) target identification and tracking. These are examples of different target detection scenarios where clothing detection enhances mission objectives [2, 78].

of the group may be too similar for positive identification, the group leader may be identifiable based on his clothing. Specifically, although their dress may appear similar to that of their subordinates, it may be constructed from textiles that present a different spectral signature. Obtaining information about the varying textiles allows for passive identification and target separation.

The second example involves an enemy combatant in the field, actively avoiding detection. Visual identification is complicated by camouflage face paint and gear, *i.e.* hoods, helmets, or gloves. In this case, the available methods of visual identification are neutralized. Other means of dismount identification, such as skin detection and facial recognition, may be complicated by the methods of concealment as well. However, by using textile identification, we can separate that individual from his surrounding environment. Because this is achieved through passive methods, it can be completed before the target is aware of detection.

The third example involves a child lost in a wooded area. In this case, the target individual would not be actively concealing himself. However, due to the nature of the terrain, traditional detection methods may be hindered. Again, using textile detection to separate that individual from the surrounding environment enables positive location and recovery.

Textile detection makes use of a fabric's spectral signature, obtained through the use of hyperspectral data. This particular type of data is collected across the electromagnetic spectrum. While this results in increased discrimination possibilities, this high dimensional data requires large amounts of time and memory to process.

Feature generation, transformations, or feature selection can alleviate the problem of too many attributes [16]. These methods reduce the number of attributes of a sample in an attempt to obtain a smaller set of highly discriminating features. Of these three, feature selection is most applicable to high dimensional data sets like hyperspectral data which can contain more than 2000 dimensions. Feature selection is used to reduce large amounts of data while preserving that data's unique classification characteristics [79]. Several feature selection methods exist, all created based on different governing principles [7, 16, 70].

The collection of hyperspectral data can be seen as an extension of multispectral imaging [9, 10]. Where multispectral only images up to 10 spectral bands and has a resolution of 10 ($\lambda/\Delta\lambda$), hyperspectral imagery collects hundreds of adjacent spectral bands with a resolution of 100. Multispectral imagers typically collect on the order of 10s of spectral channels (not necessarily contiguous) whereas hyperspectral imagers typically collect 100s of contiguous spectral channels. Therefore, hyperspectral collection produces a continuous, highly dimensional, spatially registered data set.

Highly dimensional data sets often contain redundant features that, depending on the learning algorithm, will be selected as part of the feature set [68, 93]. Redundant features can effect the accuracy and waste computational capability as they provide no new information to the class discrimination capability. Redundant features are

highly dependent and therefore the removal of a redundant feature would not change the discriminatory capability of the feature set. Guyon and Elisseeff [28] describe the general feature selection process with the addition of non-redundancy of features for data containing thousands of dimensions. Selecting non-redundant features for the feature set provides more efficient and effective class discrimination [93].

1.1 Background

This section discusses textile detection in the context of complete dismount characterization. Beginning with a brief overview of hyperspectral data and its specific application to textile detection. We explain the necessity of data reduction and its benefits and deficits. We introduce the concept of a feature selection method, especially as it relates to data reduction. The problem to be solved with this work is identified and followed by a brief description of the presented novel feature selection method.

1.2 Characteristics of Fabrics

Dismount detection combines several detection methods to increase accurate detection. Textile detection is one aspect of the complete dismount characterization. Skin detection, anthropometric data, and biometric data all provide information that can be used for identification. The AFIT/ENG Sensors Exploitation research group at WPAFB Ohio is currently developing methods for accurate modeling and detection of skin [67]. These efforts will be enhanced by modeling software like Digital Imaging and Remote Sensing Image Generation (DIRSIG) [36] created by Digital Imaging and Remote Sensing Laboratory at Rochester Institute of Technology. Analyzing specific dimensions of the body, such as weight, height, and body composition creates a database known as anthropometric data. Anthropometric data allows for the assessment in differences of body proportions of populations. Biometric data encompasses a wide range of physical characterization data. Fingerprint analysis, facial recognition, and voice recognition are all classified as biometric data. While textile classification

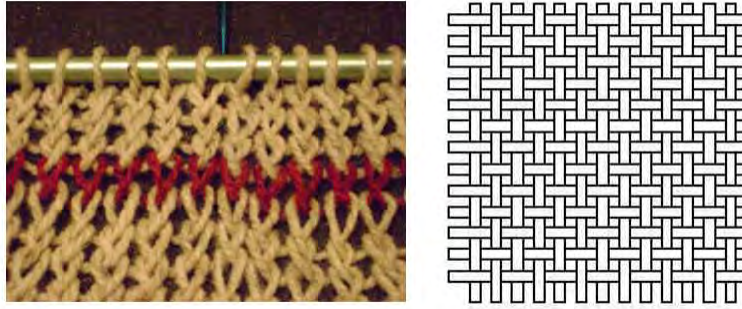


Figure 1.2: Examples of knitting (left) and weaving (right) that lead to different spectral signatures [88, 89].

is a singularly good method of detection, if it is compiled with the data obtained by the other methods listed, it will result in a complete characterization database. Therefore, the result is a significant advancement in dismount identification.

Hyperspectral skin detection modeling is heavily physics-based [1]. Because textiles have a larger variety of physical characteristics, applying the same physics-based techniques used in skin detection does not apply. Therefore, our work focuses on the textiles spectral signature that is based on the textiles characteristics. Cloth is constructed in several different ways, which results in it possessing many different physical properties. Cloth can be knitted (created with needles that pull loops through each other) or it can be woven (created using a loom), as illustrated in Fig. 1.2. Several different knitting styles and thread counts exist, each creating different physical characteristics or behaviors. Furthermore, each different type of material used in textile construction (*e.g.* cotton, nylon, wool *etc.*) has different physical properties [21, 57]. All these combinations result in a wide range of physical properties.

In the 1960s, a study was performed to characterize types of cloth [59]. The wavelength region focused on 1 to $15\mu\text{m}$ for cotton, wool, nylon, and blends of these materials. However, it is not specified in the study whether these fabrics were woven or knitted. It was concluded that there were no spectral details to differentiate between these fabrics; this is due to the physical characteristics of the fabrics, which

act as thousands of tiny blackbody radiators. It was also concluded that reflections from the surrounding surfaces (*e.g.* walls, ceiling, *etc.*) made the characterization difficult. However, an interferometer spectrometer was used to obtain the data and could account for their findings [59].

Hyperspectral data provides a more detailed look into the physical nature of fabrics, providing greater insight into their physical properties. Variations in the physical makeup of fabrics preclude the use of physics-based modeling approaches for detection and characterization. Therefore, statistical and geometric measures are incorporated into the feature selection and detection discussed in this dissertation.

1.3 Hyperspectral Data

All materials absorb, reflect, or transmit electromagnetic radiation [20]. The degree to which a material reflects electromagnetic energy is dependent on the wavelength of the energy and the material's physical characteristics. Collecting these aspects allows a material to then be spectrally identified.

Hyperspectral imagery collects the content of a material's spectral radiation over a broad spectral range, resulting in high-resolution data. Spectral information and spatial orientation of the image are recorded as they are located [22]. This is superior to normal multi-spectral collection because the information gathered contains more frequency bands, finer resolution, and wider spectral coverage. Hyperspectral data typically has small sampling intervals of about 1nm (field spectrometer [37]) to 10nm, airborne visible infrared imaging spectrometer (AVIRIS) [10], and covers the spectral range from visible through shortwave infrared, which is approximately 0.4 – 2.5 μ m [56]. Each piece of spectral data is referenced by its specific corresponding spatial location as shown in Fig. 1.3.

This ability to collect hyperspectral information allows for the identification of characteristics for different types of materials, since each image is much like that of a three-dimensional cube where the spectral bands are represented as column vectors [9].

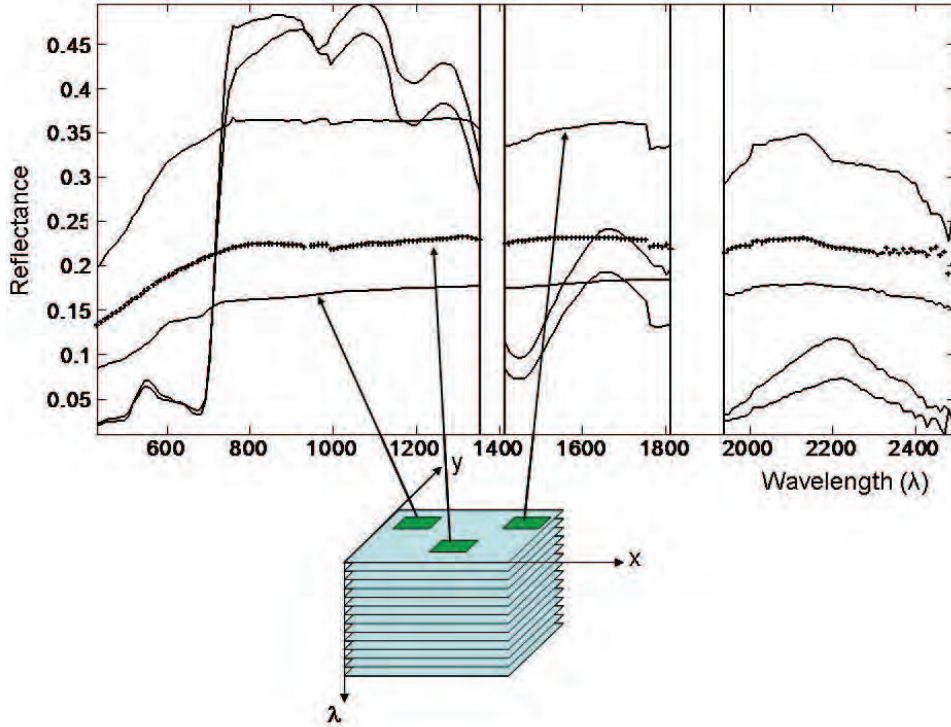


Figure 1.3: Hyperspectral data is most easily understood as an image; two dimensions define the spatial location and the third dimension is a vector of spectral values. Above is a hyperspectral image showing the different spectral information contained in hyperspectral data [65].

However, the large amounts of data generated through hyperspectral imaging can be too much information to be easily processed. Therefore, it is necessary to reduce the data, and extract only the wavelengths that assist in accurate object identification. Fig. 1.4 illustrates an instance where broad spectrum collection is necessary. The signals for different colors of the same material are shown in the figure. As can be seen in the visible region, the signals are different; however, as the wavelengths progress to shortwave infrared, they become more similar, simplifying identification. In this instance, broad spectrum collection increases the amount of information regarding each signal, allowing for more points of comparison. This approach is particularly helpful for adaptation with our method.

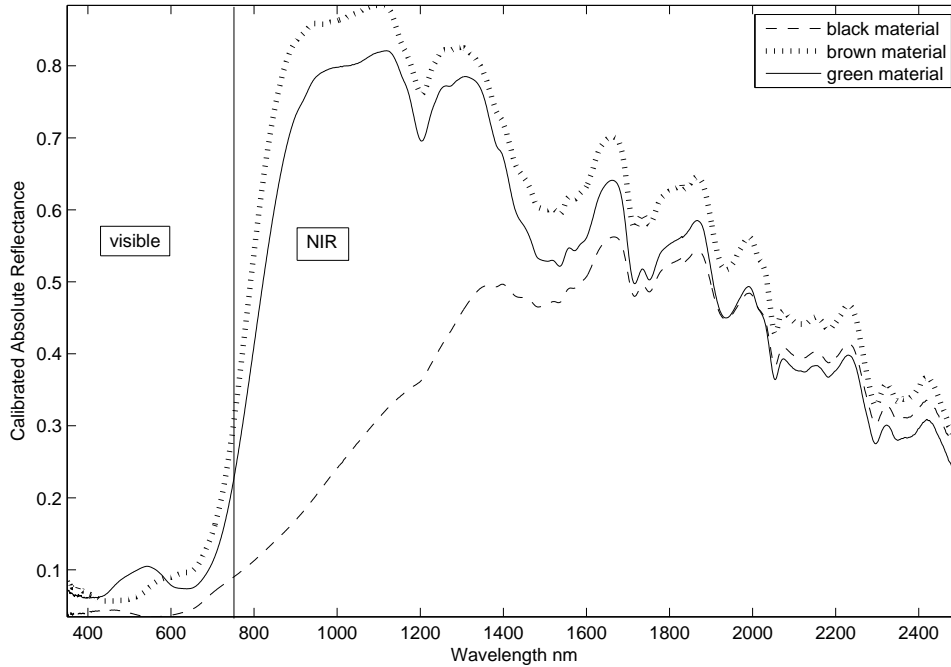


Figure 1.4: Above is an example of a textile sample, where each sample is a different color of the same material: black (dashed), brown (dotted), and green (solid). The plot exemplifies how similar the spectral characteristics are at longer wavelengths, even if the different colors make the sample dramatically different in the visible spectrum.

1.4 Feature Selection

When establishing characteristics for identification, it is helpful to have many attributes. To obtain the large data amounts necessary to characterize textiles, hyperspectral data is often used. It provides information about unique features of a particular material, information that could not be detected using multi-spectral methods and that would therefore be lost. Multi-Spectral data is high resolution data collected over relatively small bandwidths [9, 10]. Hyperspectral data is high resolution data that is collected over a continuous bandwidth and typically includes many the same regions as multi-spectral data and more. Hyperspectral imaging systems are excellent sources of information, because of their ability to collect hundreds of spectral channels. However, this may result in too much data to process in a timely or afford-

able manner. High dimensional data often contains redundant features that hinder the classification process [93]. The higher the dimensions the greater the chance of incurring the curse of dimensionality [33]. One solution is to develop a feature selection process to sift through the data and determine key features that define and discriminate classes.

A feature selection process for hyperspectral data needs to be quick, accurate, and portable. Most detection scenarios have a level of time constraint and most do not take place in the lab. Therefore, the problem of detection becomes compounded by the limits of time, computational cost and the need for portability. These requirements are complicated by the need to evaluate data signals containing noise. Noise is a significant problem in data collection; fortunately, systems are being developed with applications for field-operational collection devices [34]. These devices can efficiently and effectively collect preselected bandwidths over a wide range of wavelengths. However, collecting data in this manner requires the *a priori* selection of bands to be useful. This is because classification accuracy of a reduced feature set depends on the informational content of the selected feature set. Because the nature of hyperspectral data requires reduction or selection of bands, development of an accurate process is key. Extracting a meaningful set of features from highly correlated, highly dimensional continuous data that provides accurate discrimination of classes allows for the use of sensors that collect only the spectrum of interest, resulting in information that can be quickly processed, regardless of location. The work in this dissertation focuses on creating a novel feature selection method by applying concepts derived from multiple fields of study and adapting them for use in reducing high-dimensional data.

1.5 The Problem

What started as a relatively simple goal, accurate dismount detection, has developed into a multi-faceted complex problem.

Can we develop a method to search a highly dimensional continuous signal to provide a low-correlated small sized feature set that produces highly accurate classification results?

The rest of this dissertation provides the necessary background and information to show that our novel feature selection method solves this problem.

1.6 Methodology

To perform feature selection in highly correlated continuous data domains, random processes and non-greedy methods are extended into a novel feature selection method. Non-correlated Aided Simulated Annealing Feature Selection (NASAFS) method focuses on refining the ability to distinguish between different high correlated, high dimensional continuous data. The NASAFS method performs both random and non-greedy search aspects by using simulated annealing, which is a stochastic local search method. NASAFS incorporates a distributed spacing function and optimizes a heuristic. The distributed spacing function addresses the issue of non-redundancy by ensuring selected features are distributed throughout the data domain. The heuristic contributes to fulfilling the accuracy requirement by using distance and dependence measures to determine optimal discrimination. Feature set size is left as a user-determined variable. The results show that NASAFS produces highly accurate classification results with small feature set sizes.

1.7 Organization

In Chapter II, common mathematical procedures used in feature selection and detection are discussed. Detection techniques and search algorithms are investigated. In Chapter III, the proposed feature selection and detection methodologies are de-

scribed. The heuristic is explained, along with the search algorithm employed in the proposed feature selection method. In Chapter IV, the results of the data set analyses are shown; this is done for both hyperspectral data sets, as well as the texture data used. The novel feature selection method is compared to three common feature selection methods. Chapter V summarizes the results obtained and proposes future directions for this work.

II. Background

The goal of a feature selection method is to preserve the classification ability of the data, while reducing the feature set as much as possible to reduce computational complexity. The feature set selection process evaluates features using one of two approaches: evaluating and ranking each individual feature, or evaluating and ranking sets of features [7, 70]. Individual ranking of features can overlook redundant features and dependencies between features, which can produce inferior results. Evaluating the feature set as a set not only provides for a non-greedy feature selection, but can also more readily distinguish between relevant and redundant features [70].

In Section 2.1, three common feature selection methods are reviewed. Different theories regarding feature selection are covered in Section 2.2. Section 2.3 discusses the selection of the classifiers and feature selection methods evaluated in this work. Section 2.4 surveys current detection methods. Section 2.5 explains the governing principles and basic operations for search and detection techniques. Explanation is provided regarding mathematical foundations used to establish the statistical operations incorporated into the novel feature selection method discussed in this dissertation.

2.1 Feature Selection

The process of selecting a feature set can be difficult. There are 2^N possible solution sets, where N is the dimensionality of the data. Due to the number of possible solutions, the problem can quickly become intractable. Current feature selection methods attempt to solve this problem; however, there are many different approaches, and no one method works best for all situations. The type of feature selection method required depends on the data's size and type, as well as the intended application of the resulting feature set.

In general, feature set selection is divided into four categories: *classical*, *idealized*, *improving prediction accuracy*, and *approximating original class distribution* [16]. The classical method involves feature set selection that satisfies an established cri-

terion that defines the optimal feature set size [66]. The idealized approach finds the smallest set of features that still retains the properties necessary to accurately classify or detect a class [42]. The improving prediction accuracy method selects a feature set that is able to improve classification accuracy or reduce a cost function without decreasing that feature set's ability to accurately select features that discriminate well [47]. The goal of the approximating original class distribution method is to select the smallest possible feature set that accurately represents the complete class distribution.

Two main taxonomies for feature selection methodologies are Dash & Liu [16] and Blum & Langley [7,51]. Dash & Liu [16] classify each feature selection process by the search method employed; results are presented as a tree-like structure, which is based on the sub-categorization of the feature selection method. Blum & Langley [7, 51] use three different definitions to place all feature selection methods into three categories: *filter*, *wrapper*, and *embedded*.

Dash & Liu [16] describe all feature selection methods as having a *generation procedure*, an *evaluation function*, a *stopping criterion*, and a *validation procedure*. The generation procedure selects a set of features from all the attributes of a sample using one of three process: a *complete process*, a *heuristic process*, or a *random process*. The complete process is the most computationally intensive, but it will produce the best possible feature set for discrimination purposes. The complete method starts at the first feature and progresses through the features one feature at a time, until all features have been evaluated. With this method, there are 2^N possible solution sets; therefore, the computational burden can be great, even for small values of N . The heuristic process uses a function to determine the cost of a feature; the cost is on the order of N^2 or less, in most cases. It then provides this information to the search algorithm; this allows for the determination of the current feature's selection or rejection. Selection occurs if the current feature benefits the process; rejection occurs if the current feature degrades the process [74]. The random process selects features randomly; it then uses a function to determine if each feature should be

retained. Randomization can be beneficial to the feature selection process; however, the random process may result in a solution that is complete, but not necessarily optimal. Depending on the heuristic and search algorithm employed, the typical search space is on the order of 2^N or less [16].

The evaluation function determines the selected feature's retention based on a qualification function. Dash & Liu [16] categorize evaluation functions into five types: *distance measures*, *information measures*, *dependence measures*, *consistency measures*, and *classifier error rate measures*. Two of the five evaluation functions are discussed in further detail in this dissertation: *distance measure* and *dependence measure*. The distance measure distinguishes between classes. Euclidean distance and Mahalanobis distance are examples of distance measures. The dependence measure uses statistical information to determine a value of worth. The correlation coefficient and the covariance are examples of dependence measures.

The stopping criterion ends the search process when a predetermined number of features has been identified, or if the process itself determines that it has found an adequate number of features. The validation process is used to determine the validity of the selected features. If the proposed feature set is an accurate enough representation of the complete data set, then it passes the validation criteria.

Blum & Langley's feature selection taxonomy divides feature selection methods into three groups: *filters*, *wrappers*, or *embedded* [7, 51]. The filter method uses a training set and takes into account certain properties of the data. Those properties then help to determine which set of features will be used in the detection algorithm; however, no learning takes place in the selection of these features [7, 50, 51, 54]. Examples of filter methods are maximum entropy, measures of statistical redundancy, ReliefF [48], Bhattacharyya [5], and linear dependence [44, 90]. The *wrapper* method picks a set of candidate features, and then uses those features with the training data and a specific machine learning process in order to determine the accuracy of the candidate feature set. The machine learning process is not specific to the selection process

and can be interchanged with other learning processes. Features are combined and evaluated until an appropriate feature set is detected [7,50,51,54]. Examples of wrapper methods include breadth-first-search, genetic algorithms, and simulated annealing searches [74]. Embedded methods determine a feature's *goodness* as it selects the feature set. The feature set continually updates in order to produce a more desirable feature set. This process continues until the stopping criterion is met [16]. In other words, the embedded method adds or subtracts features in response to prediction errors. The selection and learning process are integrated and can not be separated [50]; this is thought to produce a more discriminatory feature set. Examples of embedded methods include C4.5 and the generalized relevance learning vector quantization (GRLVQ) family of classifiers [16,31,61].

2.2 Current Methods for Feature Selection

This section covers several different methodologies that can be adopted for feature selection techniques. The methods discussed include: Principal Component Analysis (PCA), Relief/ReliefF, GRLVQ, and Bhattacharyya.

PCA [6,9,10,41] is a technique used in many applications. PCA operates in an transformed space and computes the data's eigenvectors and eigenvalues from the covariance matrix. The number of eigenvectors kept is determined by the user. The total number of eigenvectors is on the order of the dimensionality of the data. Typically, only a small group of the eigenvectors are necessary to reduce the number of dimensions in the data and maintain classification accuracy. The eigenvectors are chosen in descending order of their eigenvalues. These eigenvectors form the new basis for the data set, and allow the data to be represented by a different set of axes. Depending on the data, this results in a better discriminating class separation. Because PCA's discriminatory capability is based on the linear combinations of the features and is projected into the transformed space, determining a feature in the original data domain that has the greatest discriminatory capability is subjective [8]. The resulting feature set is often found to be highly correlated and redundant. Therefore, PCA is

an inadequate class discrimination method for high-dimensional data, noisy data, or data with missing parts.

Relief [43] uses a distance measure to rank each feature of a sample to its nearest in-class and out-of-class sample; the distance is the weighting of the feature, which is then used to determine the best discriminating features. This technique was developed by Kira and Rendell for the two class problem [42, 43]. It was later adapted to the multi-class problem by Kononenko [48], and is ReliefF. However, hyperspectral data has features that are highly correlated and dependent; therefore, ReliefF produces a feature set that is highly correlated and redundant (see e.g., González [38]).

The Bhattacharyya distance [5, 24] is used as a method of feature selection by means of a measure of the overlap of probability density functions (*pdf*). The *pdf* of each attribute between all class pairs ($C(C-1)/2$) is used in the Bhattacharyya function. Each attribute is then weighted based on the mean, median, or minimum [25] values for each attribute compared per class. The minimum surface Bhattacharyya feature selection method presented in [25] is a relatively good feature selection method for the multi-class problem. However, the method and its deviations do not make provisions to prevent highly correlated or redundant feature sets.

GRLVQ [31] and GRLVQI [45, 60, 61] are large margin classifiers and use an adaptive diagonal metric (feature weighting scheme) to provide feature ranking information based on an attribute's discriminatory capability. These techniques are a continuation of LVQ2.1 [28, 46], which moves prototype vectors around in data space to determine a decision boundary for classification. Prototype updates are based on a differential shifting strategy that helps refine the decision boundary. The selected features are determined based on which attributes have the greatest contribution to the classification. In order to obtain feature sets that are not highly correlated, and to reduce the total number of features in the feature set, transformation of the data prior to the GRLVQI process has been suggested [60].

2.2.1 Other Specific Related Works. In Section 2.1, the feature selection methodologies discussed use different techniques to select a feature set that provides discrimination capability. These methods span the different taxonomies, and each has multiple variations; there are also other feature selection methods that use several combinations of different methods. Some of the works most relevant to our method are discussed in the following paragraphs.

The work of Kumar, *et al.* [49] states that most hyperspectral feature extraction methodologies ignore the ordering information between adjacent bands, and typically produce a global feature set, which is an inefficient utilization of the data. Kumar *et al.* propose a pairwise method that merges adjacent band subsets, in order to obtain a small number of discriminatory features. They base their feature selection methodology on Coefman and Wickerhauser [13], and Saito and Coifman [75], who divide a hyperspectral signal into wavelet packets and introduce a classification scheme called local discriminant bases (LDB). The LDB method projects the signal onto orthonormal bases and then either minimizes or maximizes an entropy/cross-entropy cost function. The LDB method then divides the hyperspectral signal into segments of equal lengths. LDB evaluates the first two attributes and determines individual and combined relevance, based on an entropy cost function. Depending on the goal of the cost function, the attributes are either combined or left as individuals. It then moves to the next two attributes and performs the same operation. It continues this process in a bottom-up approach on a signal decomposed by the discrete wavelet transform [13,75]. Kumar *et al.* [49] propose an alteration to the LDB method that does not divide the signal into equal segments; instead they propose unequal divisions to provide better classification using a three step process: recursive partitioning of adjacent bands into non-overlapping groups, merging bands within each group (linear combination), and a selection of band groupings based on the best discrimination between classes.

Peng *et al.* [68] present a feature selection method based on mutual information that capitalizes on maximum-dependency and minimum-redundancy. Peng *et al.* also

state that a good feature set is not necessarily a set of individually good features. They propose that the *goodness* of the feature set is instead dependent on the feature set as a whole; this is also proposed by Jain *et al.* [39], and Cover and Thomas [14, 15]. Max-dependency, as stated in Peng *et al.* [68], is a mutual information process used to determine the features with the largest dependency on the target class. However, since this is difficult with high-dimensional data, they use max-relevance. Max-relevance incorporates mutual information; however, it can be highly redundant. To alleviate this tendency, they combine max-relevance with a min-redundancy from their previous work that also uses mutual information [19]. Peng *et al.* [68] show that their previous work (minimal-redundancy-maximal-relevance (mRMR)) produces the same results as max-dependency for the sequentially added feature approach. However, they suggest a better approach is a two-fold process, using mRMR initially, then using a more sophisticated search method to further reduce the feature set.

Other techniques use a strategy that reduces the dimensionality of the data by grouping attributes based on a correlation matrix to determine their correlation to each other, then applying a type of feature selection method [79]. Some feature selection methods use a stochastic function as a preprocessing technique to feed a feature validation process, as in the work by Pizzi *et al.* [69]. Pizzi *et al.* use a probabilistic neural network to determine if a randomly selected set of features is a good discriminating feature set.

While some of the methods discussed in this section attempt to produce non-redundant feature sets, their methods differ significantly from the novel feature selection method presented in this work. However, all the feature selection methods evaluated against our novel feature selection method tend to produce highly correlated feature sets, indicating redundant features. A robust feature set should not only provide accurate classification, but also be non-redundant to ensure class discrimination in noisy environments or instances of data occlusion.

2.3 Classifier and Feature Selection Method Qualifications

Holmstrom *et al.* [35] categorize classifiers according to their neural net qualities. These qualities are plotted in a two-dimensional graph, where the axes represent design complexity. These design complexities are the flexibility of a classifier’s architecture with respect to its discriminant function (horizontal axis), and the classifier’s learning ability with respect to its neural net training (vertical axis) [35]. Fig. 2.1 is the schematic of the neural characteristics of some classifiers. Some are originally determined by Holmstrom *et al.* [35]; others have been added by González [38], based on Holmstrom *et al.* definitions. In this diagram, C4.5 falls into the upper right quadrant, indicating an adequate amount of complexity and some neural net capability. However, Naïve Bayes falls into the lower left quadrant. This is because Naïve Bayes has little neural net capability, due to the fact that it is predominantly a statistical learning machine; it has little complexity to its discrimination function, as well. The classifiers represented in this work are purposefully chosen to span the Holmstrom taxonomy domain.

The feature set selection methods chosen for comparison with our novel feature selection method belong to different categories, as defined by both Dash & Liu [16] and Blum & Langley [7, 51]. The Dash & Liu [16] taxonomy identifies ReliefF as a heuristic based search method; according to Blum & Langley [7, 51], it is a filter method. GRLVQI and Bhattacharyya are both classified by Blum & Langley [7, 51]; however, GRLVQI is an embedded method, whereas Bhattacharyya is a filter method. While these feature selection methods do not cover all the taxonomy categories, this is a representative selection of them.

2.4 Target Detection

Both our novel feature method and our detection algorithm exploit the structure of hyperspectral data. Additionally, since our parameters are discrete, the detection algorithm employed is *simple hypothesis testing* [58]. Hypothesis testing of this type can be composite or binary; for this work, a binary approach is taken. To model this

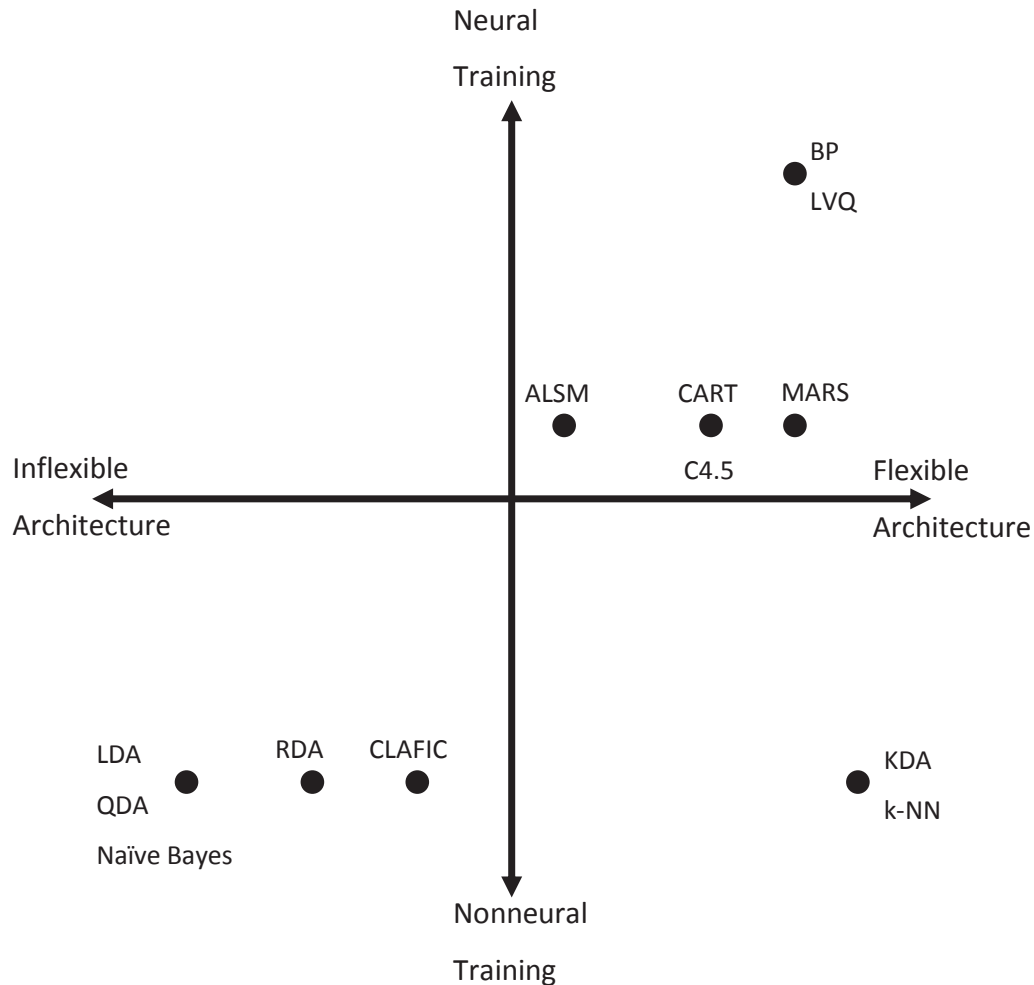


Figure 2.1: Schematic of the classifiers, based on their neural net categorization, as determined by Holmstrom *et al.* [35], and added to by González [38].

approach, consider an example of attempting to detect the presence of an airplane in a specific area of airspace. Let H_0 be the hypothesis that the plane is not in the airspace, and let H_1 be the hypothesis that the plane is in the airspace. The raw received radar return of the signal only (no noise) is considered the observation data. If the signal is large enough, then we conclude hypothesis H_1 ; if not, we conclude H_0 . The trick is determining the threshold at which H_1 can be concluded; in this example, this is determining where the threshold is in relation to the magnitude of the signal return.

Using this understanding of the binary detection method, we can define some statistical variables. H_0 is called the *null* hypothesis. This is the expected result; for our example, this is the plane is not present. H_1 is the alternative hypothesis, the unknown condition that is possible, but not expected; in our example, this would be the plane is present. Therefore, using the above terminology and the previous example, $P[H_1|H_0]$ is the probability of thinking there is a plane present, given that there is no plane present; this is referred to as the probability of false alarm, denoted as P_F . $P[H_1|H_1]$ is the probability of thinking there is a plane present, given there is a plane present; this is referred to as the probability of detection, denoted as P_D . $P[H_0|H_1]$ is the probability of thinking there is not a plane present, given there is a plane present; this is referred to as the probability of a miss, denoted as P_M , where $P_M = 1 - P_D$. $P[H_0|H_0]$ is the probability of thinking there is not a plane present, given there is not a plane present; this is referred to as the probability of a rejection, denoted as P_R , where $P_R = 1 - P_F$ [58, 76]. The understanding then is that the prediction is either right ($P[H_1|H_1]$, $P[H_0|H_0]$) or it is wrong ($P[H_0|H_1]$, $P[H_1|H_0]$).

A Receiver Operating Characteristic (ROC) curve can be used for binary classification systems to characterize detection capabilities for either singular or multiple systems. A ROC curve is a graph displaying P_D versus P_F , as the threshold is varied. This allows for an accuracy measurement of a certain system to be displayed, with regard to the degree of how inaccurate it can be. For example, Fig. 2.2 shows a generic ROC curve; using this diagram, it can be determined if a P_F of 0.2 is desired, then the system will produce a $P_D = 0.537$. It is also shown that with a greater probability of detection comes a greater probability of false alarms. Ultimately, it is desirable for a system to have a high probability of detection and a low probability of false alarm.

Bayes' rule is a way to determine the *a posteriori* probability given the *a priori* probability, and is often used in detection systems. Bayes' rule is defined as:

$$P[A|B] = \frac{P[B|A]P[A]}{P[B]}. \quad (2.1)$$

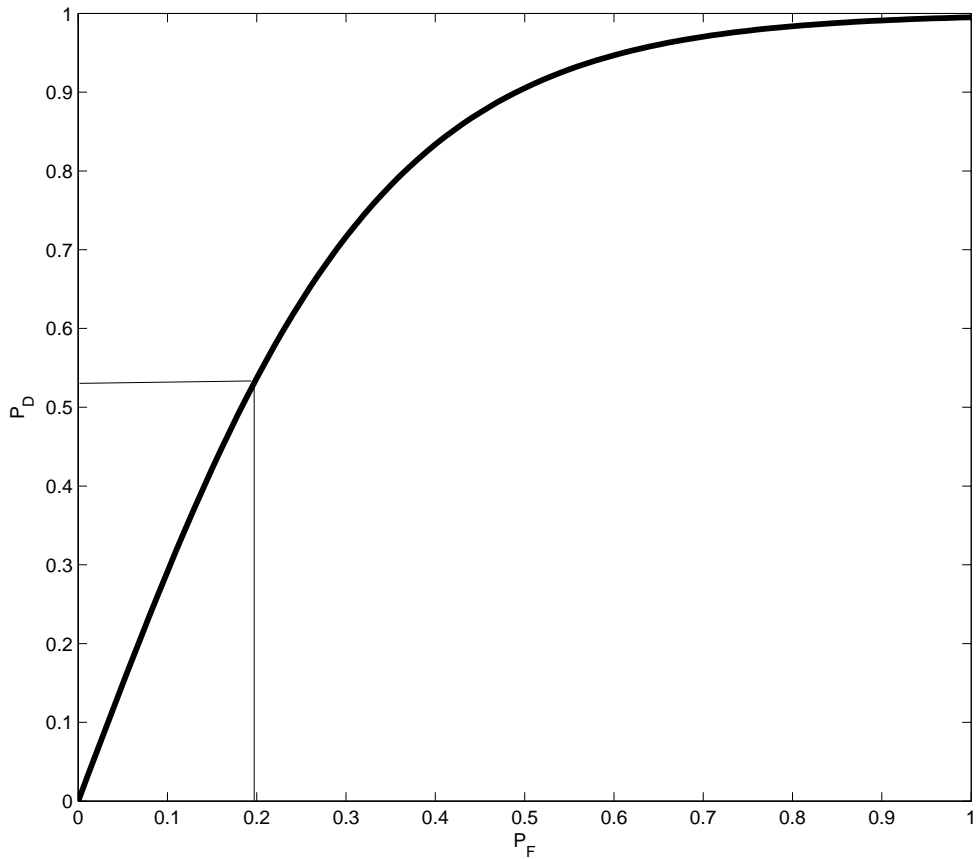


Figure 2.2: Example of a generic Receiver Operating Characteristic (ROC) curve.

This is read as: the probability of A given B is equal to the probability of B given A multiplied by the probability of A, and then divided by the probability of B.

Two common detection systems are Bayes' test/likelihood ratio (BLR) and Minimax. For our airplane example, the BLR is a ratio of the probability density function (*pdf*) of the case where the plane exists in airspace versus the *pdf* of the case where the plane does not exist in the airspace. This ratio is compared to a cost function, which then determines if it is the null hypothesis or the alternative hypothesis that is chosen. The BLR is used to determine a threshold for defining target present or target absent. The cost function is a ratio of the prior probabilities multiplied by the

ratio of the cost of each possible case; therefore, the BLR is derived from the risk:

$$Risk = E[cost] = \sum_{i,j} C_{ij}P[H_i|H_j]P_j \quad (2.2)$$

where $E[\cdot]$ is the expectation, i and j are binary (*i.e.* 0 or 1) and C_{00} is the cost of guessing H_0 when H_0 is true, C_{10} is the cost of guessing H_1 when H_0 is true, C_{01} is the cost of guessing H_0 when H_1 is true, and C_{11} is the cost of guessing H_1 when H_1 is true. If $p_0(x) = p(x|H_0)$ and $p_1(x) = p(x|H_1)$, and $\int_{S_0} p_0(x)dx$ is the total probability of picking H_0 given H_0 is true, and since

$$\int_{S_0} p(x)dx + \int_{S_1} p(x)dx = \int_S p_0(x)dx = 1 \quad (2.3)$$

where S_0 is the decision region of H_0 , and S_1 is the decision region of H_1 , and S is the total region ($S_0 + S_1$), then:

$$\int_{S_1} p(x)dx = 1 - \int_{S_0} p(x)dx. \quad (2.4)$$

Therefore,

$$\begin{aligned} Risk &= C_{00}P_0 \int_{S_0} p_0(x)dx + const1(1) - C_{10}P_0 \int_{S_0} p_0(x)dx + const2(1) \\ &\quad - C_{11}P_1 \int_{S_0} p_1(x)dx + C_{01}P_1 \int_{S_0} p_1(x)dx \\ &= const1 + const2 + \int_{S_0} [C_{00}P_0p_0(x) - C_{10}P_0p_0(x) - \\ &\quad C_{11}P_1p_1(x) + C_{01}P_1p_1(x)]dx \end{aligned} \quad (2.5)$$

where $const(1) = C_{10}P_0$ and $const(2) = C_{11}P_1$. Since the goal is to minimize risk, $const1$ and $const2$ can be ignored, because we have no control over these constants. It can be seen from Eqn. 2.5 that to minimize the risk, the equations within the

summation need to be as small as possible (actually, less than zero)

$$(C_{00}P_0 - C_{10}P_0)(p_0(x)) - (C_{11}P_1 - C_{01}P_1)(p_1(x)) < 0. \quad (2.6)$$

Eqn 2.6 can be rewritten to see the form of the BLR as:

$$(C_{00} - C_{10})P_0(p_0(x)) < (C_{11} - C_{01})P_1(p_1(x)). \quad (2.7)$$

Cross multiplying and solving for the pdf's yields,

$$\frac{(C_{00} - C_{10})P_0}{(C_{11} - C_{01})P_1} > \frac{p_1(x)}{p_0(x)}, \quad (2.8)$$

where the inequality changes due to the fact that the cost $C_{01} > C_{11}$. Therefore, the BLR is written as:

$$\Lambda(x) = \frac{p_1(x)}{p_0(x)} \underset{< H_0}{\overset{> H_1}{\left(\frac{C_{00} - C_{10}}{C_{11} - C_{01}} \right)}} \cdot \frac{P_0}{P_1} \quad (2.9)$$

where $p_1(x)$ and $p_0(x)$ are *pdfs*, and P_0 and P_1 are the prior probabilities of each hypothesis (H_0 and H_1 , respectively).

The Minimax forms a decision rule in order to minimize the risk for the worst case of P_0 , when P_0 is unknown. For the Minimax, first the Bayes' test is solved in terms of the unknown P_0 , then the risk is evaluated assuming P_0 is correct. The risk is defined in Eqn. 2.2 [58]. The next step is to find the argument that makes $R(P_0)$ peak; BLR is evaluated at that point, and $R(P_0)$ is used to determine the threshold.

The detection algorithm proposed in this paper uses a threshold; however, that threshold is determined with statistical measures, not the usual techniques as found in the BLR. The detector presented in this work is discussed in detail in Chapter III. The statistics used in conjunction with the detector of this work are discussed in the following section.

2.5 Search Algorithms and Statistics

The definition of *search*, in terms of algorithms and according to Russel and Norvig [74], is the process of looking for the best possible sequence, given several choices that will achieve the goal state. A search algorithm can be divided into several different categories; all are based on the intended use of this definition of search.

The principles that are used to create a search algorithm are called *agent programs* [74]. There are four basic types of agent programs: *simple reflex agents*, *model-based reflex agents*, *goal-based agents*, and *utility-based agents*.

A simple reflex agent is a process in which decisions are based on the current environment [74]. These types of agents do not use information about past decisions or environments. They also do not store information about the environment; decisions are based solely on currently observable information. Therefore, they are of little use when the current situation is not fully observable.

Model-based reflex agents also base decisions on the present environment; however, they differ from simple reflex agents in their use of information stored regarding unobservable parts of the environment [74]. This information formulates a working model of the environment that assists in decision-making; this is especially useful in situations where the decision must be made based on partial observability.

Goal-based agents make decisions based on both current and modeled information [74]. However, the current information and the stored model of the environment are combined in order to achieve a particular goal. Decisions are then made according to which choice will lead to the desired goal state.

Utility-based agents incorporate all of the aspects of the other three agent programs. They differ from goal-based agents because while they base decisions on the attainment of a goal, they evaluate the possible routes to a goal and choose the most efficient route possible [74].

These search algorithms receive a problem and return a solution. In some cases, the algorithm does not receive information about the problem, only the problem's

definition [74]. These algorithms are known as uniformed algorithms. In contrast, an informed algorithm receives not only the definition of the problem, but also information about the problem; this information helps the algorithm to more efficiently determine the direction of its search.

The performance of an agent is determined by its success at finding desirable solutions; the criteria used to establish this are called performance measures. These performance measures are: *completeness*, *optimality*, *time complexity*, and *space complexity* [74]. Completeness is the ability of the search method to find a solution, given that there is a solution. Optimality is determined by the path cost function, where the optimal solution is the solution with the lowest path cost. Time complexity refers to time required by the algorithm for the search and space complexity is determined by the amount of memory required to perform the search.

For a better understanding of some of these definitions, a few examples are given. An uninformed search, that is complete, that has a time complexity of $O(b^{d+1})$ (where b is the branching factor, and d is the depth of the shallowest solution), that has a space complexity of $O(b^{d+1})$, and that is optimal is a breadth-first-search. When executed, a breadth-first-search expands each subsequent node from the root node. If none of the explored nodes are the solution, the search goes to the first subsequent node created and expands all of its subsequent nodes and checks for the solution. If no solution is found, the previous nodes are retained in memory and the search proceeds to the second subsequent node; the process of expansion is repeated until every node is expanded or the goal is found. The breadth-first-search method is complete, but it is costly in terms of time and memory requirements [16, 74].

An informed search can be incomplete and not always optimal, but have a space and time complexity of $O(b^m)$ (where b is the branching factor and m is the maximum depth of the search space). An example of an informed search is the greedy-best-first-search [74]. The greedy-best-first-search operates much like the breadth-first-search; however, after the initial expansion of the root node, each subsequent node has a value

attached to it. This value is the estimate of the time required by the search algorithm to achieve the goal if that particular node is used to determine the solution. In this instance, the solution will be the node with the best assigned value. The function that provides this value is called a heuristic. However, completeness and optimality of a search algorithm are dependent on how well the heuristic performs for that particular situation.

An informed search that is complete *and* optimal is the A^* search. This search determines an actual cost for reaching the subsequent node, rather than estimating that cost. This cost is then combined with the use of a heuristic. This is done in order to determine the best node for use in subsequent expansion [74].

These search methods help to explain the basic concepts of search algorithms. However, the type of search used in this work requires the exploration of other methods. In our case, the actual path to the goal does not matter, only the fact that the goal is obtained; therefore, a local search may be employed. In a local search, the algorithm operates using its current state. It then moves to neighboring states of the current state; this is based on the heuristic value assigned to its neighbors [74]. Local searches use minimal amounts of memory, and are useful in large or continuous spaces; these factors make it well-suited for feature selection in hyperspectral data, as well as for use in optimization problems. The goal of a local search is to find the maximum or minimum of the cost function of a data set; ideally, this is a global maximum or minimum, not a local maximum or minimum.

Hill-climbing is a local search algorithm that chooses, in a greedy manner, the neighbor that leads it in the direction of the nearest minimum or maximum. The problem with a hill-climbing algorithm is that it can get stuck in a local minimum or maximum and fail to find the optimal goal. Our novel feature selection method employs simulated annealing search (SA). This is an informed stochastic hill-climbing local search that adds randomness to the algorithm; this is done in an attempt to prevent the hill-climbing method from getting caught in a local maximum or minimum.

SA achieves this randomness by combining the hill-climbing technique with a random walk [74]. It is modeled after the annealing process in metallurgy, which involves gradually cooling down metal, allowing the material to change its crystalline structure, thereby producing a stronger, more efficient material. The SA process starts with an initial state that is chosen at random. A value is assigned, and the process continues to the next state. These states are then compared, and the state that improves the current situation is kept. It is possible that a state that does not improve the current situation may be kept; this is due to the random nature of the process. It uses the delta by which it worsens the state and modulates it with an exponential function, $(\exp(\frac{err}{T}))$, which is a function of the cooling temperature T . If the difference of the randomly chosen state decreases the current value, the state has a chance of being kept; however, this is at a probability less than one [74]. From the above exponential expression, we can see the probability lessens as the negative difference increases in negativity, and as T decreases. The temperature T is set to decrease according to a predefined decay function. Therefore, as time increases, the possibility of accepting a state with a value less than the current state decreases to zero. The SA process is illustrated in Fig. 2.3.

SA is considered a wrapper feature selection search method based on the Blum & Langley taxonomy [7, 44, 51]. However, the feature selection method proposed in this dissertation is not technically a wrapper method. The heuristic incorporated in the algorithm contains filter methodologies (*i.e.* dependence measures and statistical calculations [90]) and gives the overall connotation of an embedded method. Therefore, it can be concluded that the feature selection methodology set forth in this work is a hybrid that most closely resembles an embedded methodology. Using the Dash & Liu [16] taxonomy, it is again difficult to determine the exact classification. The incorporation of a heuristic implies its classification to be a heuristic process; however, simulated annealing is a random process. Therefore, our novel feature selection method is considered to be a hybrid according to the Dash & Liu [16] taxonomy as well.

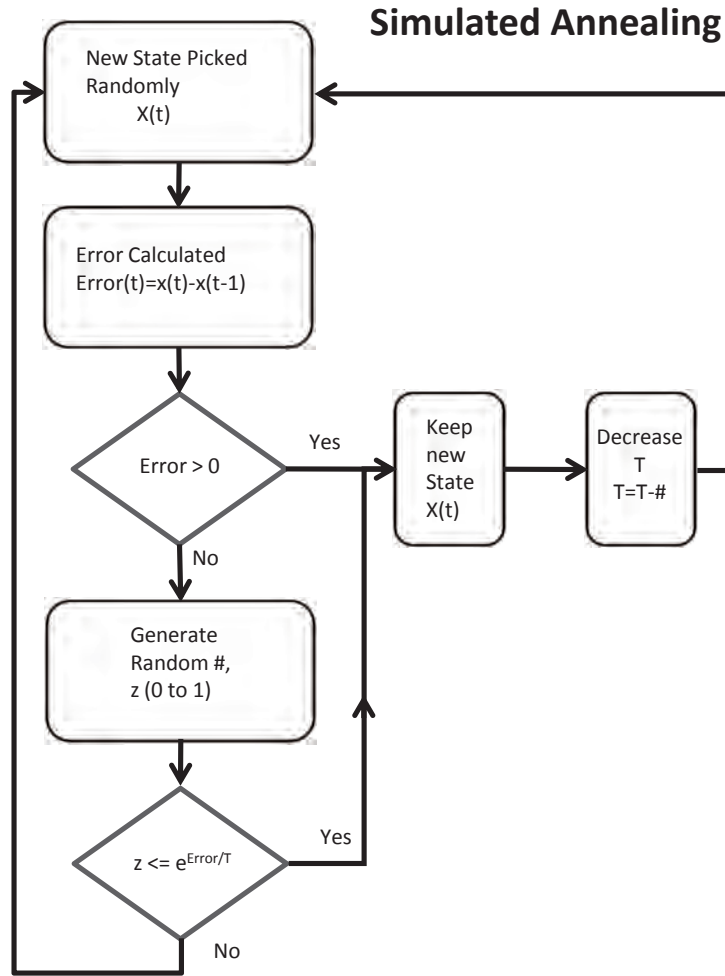


Figure 2.3: Flow chart for the simulated annealing search process.

As previously stated, the heuristic used in this work contains dependence measures and statistical calculations. These are common and excellent methods of delineating between random data sets, especially since this work deals with random processes. Some of the common geometrical and statistical measures used are the Euclidean distance metric and covariance and correlation dependence statistics. A random process (RP) is considered to be a sequence of random variables (RV). The Euclidean distance between two random vectors $\mathbf{X} = (X_1, X_2, \dots, X_k)$ and $\mathbf{Y} = (Y_1, Y_2, \dots, Y_k)$ defined on $\Omega_{\mathbf{X}, \mathbf{Y}}$ with outcomes $(x_i, y_i : i = 1, \dots, k) \in \Omega$ according

to distribution $F_{\mathbf{X}, \mathbf{Y}}$. The Euclidean distance is therefore,

$$\begin{aligned} Distance(\mathbf{X}, \mathbf{Y}) &= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_k - y_k)^2} \\ &= \sqrt{\sum_{i=1}^k (x_i - y_i)^2}. \end{aligned} \tag{2.10}$$

Which is complicated but can be overcome by using the more simplified Manhattan distance given by:

$$Distance(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^k |(x_i - y_i)| \tag{2.11}$$

where $|(\cdot)|$ is the absolute value.

Covariance and correlation are dependence measures, which can best be understood by first defining variance. Variance, σ^2 , is a measure of data distribution around the mean (average value of a given set of data) [52, 58], defined as:

$$\sigma^2(X) = E[(X - E[X])^2] \tag{2.12}$$

where $E[\cdot]$ is the expectation. Covariance, cov , is the measure of how two variables (RVs in this case) change in relation to each other. In the case where it is random processes that are measured rather than random variables, the term cross-covariance, $xcov$, is used. A positive covariance signifies that the rates of change for the two variables are equivalent. However, if the covariance is zero, then the variables are determined to be uncorrelated. Correlation is defined as the linear dependence, or lack thereof, between two variables. Covariance and correlation are shown in Eqn. 2.13 and Eqn. 2.14 as shown below [52, 58].

$$\begin{aligned} Cov(x, y) &= E[(x - E[x])(y - E[y])] \\ &= r(x, y) - E[x]E[y] \end{aligned} \tag{2.13}$$

where,

$$r(x, y) = E[xy], \quad (2.14)$$

where r is the correlation function, and $E[\cdot]$ is the expectation.

The autocovariance is the covariance of an RV with itself time shifted, where $Cov_X(t_1, t_2) = Cov(X(t_1), X(t_2))$ [58]. The autocorrelation, R , is the same concept as the autocovariance, written as: $R_X(t_1, t_2) = E[X(t_1), X(t_2)]$ [58].

2.6 Redundancy in Feature Sets

Feature set redundancy is a known problem that few have attempted to solve. Sometimes this redundancy can be observed by viewing the data signal in correlation with the chosen feature sets, as shown in Fig. 2.4. This figure shows the locations of the six features for the feature set chosen by the Bhattacharyya method. However, the visual inspection method is not computationally viable for determining redundancy in a feature selection system; to accurately ascertain redundancy, another method must be used. According to Hall and Smith [30], a good heuristic is one that selects features that are not correlated with each other, but that are highly correlated with their associated class. Therefore, a heuristic is one method that can be used to eliminate redundancy. Statistical methods are also commonly used to indicate feature set redundancy has occurred or will occur. The correlation coefficient is the most common statistic used to determine probable redundancy. In the paper by the current authors, Clark *et al.* [11], the spectral domain is distributively divided to obtain non-redundant feature sets. The procedure in Clark *et al.* [11] divides the spectral domain into equal regions; a spacing function is then applied, based on these regions. A better division of the spectral domain can be achieved by using the correlation matrix of the data, and provides feature sets with lower correlation. A current method by Punitha and Santhanam [70] attempts to solve the redundancy issue by using PCA to select features that are relevant. These features are then subjected to the correlation function to determine redundancy. However, the use of PCA has several drawbacks,

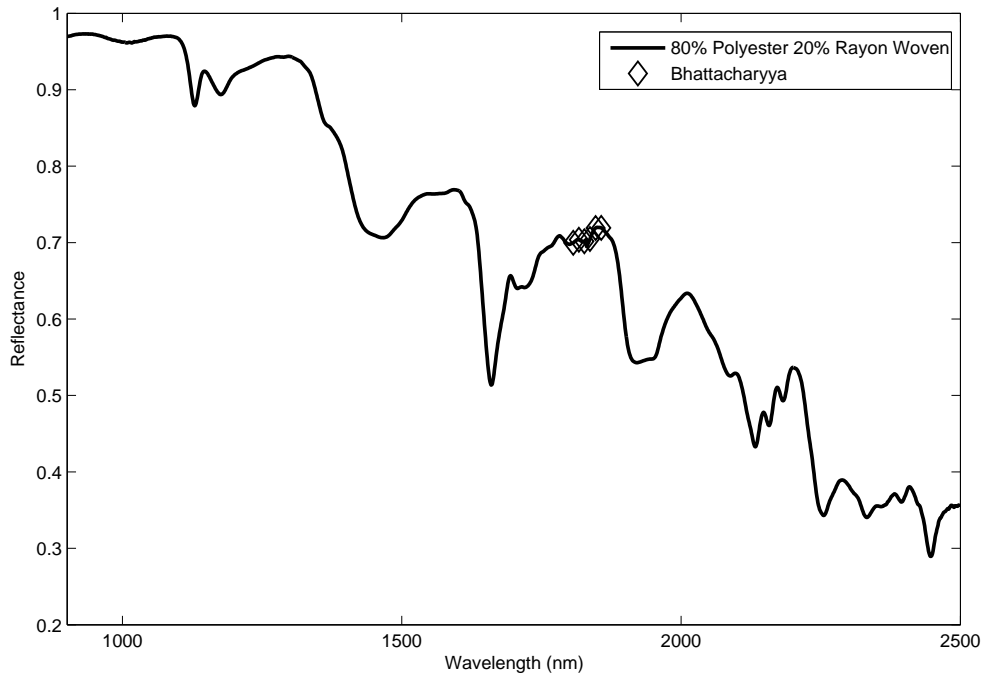


Figure 2.4: Example of the redundancy produced by the Bhattacharyya feature set selection method. The diamonds are the feature locations of the six features chosen by the Bhattacharyya method, shown on the textile signal 80% Polyester 20% Rayon.

as discussed in Section 2.2. Overall, this method produces a smaller feature set than the other methods tested, and it is less efficient as well [70]. Hall and Smith [29, 30] present a correlation-based feature selection method, in which a heuristic evaluates the relevance and redundancy of features; this is based on a best-first-search method with a stopping criterion. This method is comparable to, if not better than, other wrapper methods. However, no other categories of feature selection methods are evaluated, and there is no comparison provided regarding correlation of the feature sets obtained. Mutual information is another approach to solving redundancy issues, as presented in Ding and Peng [19]. Their method incorporates evaluation of features on an individual basis. Alternatively, Hastie *et al.* [32] express that it is best to handle redundant features as a feature set, where relevance and redundancy are determined *simultaneously*. Therefore, they propose a method where redundant features are handled explicitly. This is similar to the basic premise of the novel feature selection

method presented in this work; however, our method reverses the order suggested by Hastie *et al.* [32]. The method used in Hastie *et al.* evaluates each feature for relevance using a feature's correlation value to its class. The redundancy is then evaluated in a pairwise manner with another feature in the feature set. Qu *et al.* propose a feature selection method that uses a forward selection hill-climbing search. Irrelevant features are removed from the ranked feature list by a mutual information scheme and redundant features are then removed via a pairwise decision correlation process [72]. In these methods, redundancy is acknowledged to be a problem, and is determined by some statistical evaluation of the individual features. Our novel feature selection method uses the correlation matrix to determine redundancy, but not in the same manner as the previously mentioned works (described fully in Section 3.1.1.2).

2.7 Summary

This chapter provides a basic understanding of the feature selection process using two prominent taxonomies. The processes of several common feature selection methodologies are discussed, as are those of a few new methods. The feature selection methods chosen for comparison in this work are shown to be a good representation of the taxonomies discussed. The operating principles of detection and classification systems are outlined. The various types of search algorithms are reviewed, and the specific type of search method used in this work is covered. Several statistical measures are explained to provide a better understanding of the methods incorporated in the novel feature selection method presented in this work. The problem of feature set redundancy is reviewed, and several current methods developed to alleviate this problem are explained. The literature review and background knowledge provided determine that an accurate feature selection method, using a stochastic search algorithm to produce a non-redundant feature set, has not been achieved.

III. Feature Selection and Detection

The typical feature selection process incorporates a couple of techniques. A feature selection method and a function to determine feature goodness. The feature selection method consists of a search technique, where the goodness function is typically some sort of a heuristic, where goodness provides an indication of classification accuracy. Selecting features of a feature set that have low correlation are desired because correlation indicates redundancy [18, 93]. Redundant features typically do not add new information to the discriminating capability of the feature set, and are generally considered unnecessary [28]. Additionally, a feature set with redundant features wastes a collection system's computational resources.

The following sections discuss two versions of our novel feature selection method that provides non-redundant features. The first version, Non-correlated Aided Simulated Annealing Feature Selection (NASAFS), uses a correlation matrix and produces rigid sub-regions of the data domain to assist feature non-redundancy. In this version, the distributed spacing technique determines low correlations of features (non-redundancy) and operates as a cross-check outside of the heuristic method. The second version, Non-correlated Aided Simulated Annealing Feature Selection - Integrated Distribution Function (NASAFS-IDF), improves the first by eliminating the correlation matrix and the use of data sub-regions. This version optimizes the low correlation of features by incorporating it into the heuristic. The second version also differs from the first in its use of a one-versus-all approach, rather than a class pairwise process. In Sections 3.1 and 3.2, a step-by-step description of our novel feature selection method and the improved version is given. Sections 3.1.1 and 3.2.1 detail the steps involved in the novel feature selection method and its improved version. Sections 3.1.2 and 3.2.2 describe the feature database and how it is created. Finally, Section 3.3 discusses the operation of the detector developed for this work.

3.1 Non-correlated Aided Simulated Annealing Feature Selection

Overview

The first version feature selection method produces low correlated feature sets using a stochastic search technique that is aided by a heuristic. This method is called the Non-correlated Aided Simulated Annealing Feature Selection (NASAFS) method. NASAFS selects features from the collected data domain, with low correlation and good classification accuracy. Attributes of the data domain contain discrimination-rich information, as it directly relates to each feature. However, this information can be changed or degraded when transformed, which leads to sub-standard feature selections. Therefore, feature selection methods that operate in the data domain have an advantage to those that transform the data prior to feature selection.

NASAFS accomplishes the low correlation by using a distributed spacing technique adapted from multi-objective optimization problems to constrain the search process [12, 85]. This spacing technique determines the spread of a feature set's features across the domain by calculating the normalized difference of the expected spread of features to the actual spread of features. This ensures a set of features that has low correlation, which in turn produces a feature set that has highly robust in discriminatory capability; due to the decreased redundancy in the information content. The feature set is found using simulated annealing (SA), a the hill-climbing stochastic search method [74].

SA uses a heuristic during the search to decrease the computational requirements of the search algorithm, the feature selection search is accomplished in a class pairwise manner. Each pair of features are combined to create a database of *distinguishable features*, which is used to discriminate the reference class from all other classes. The feature selection process builds a small database of *distinguishable features* in order to discriminate the reference class from all other classes. This database is then used to create a feature set with a highly accurate detection rate. From these histograms

a single feature set for each class is created. The resulting feature set is then used by a classification method to categorize new samples into one of the given classes.

Let x_j represent a bin, where $x_j \subseteq X_i$ and $Y = \{x_1, x_2, x_3, \dots, x_m\}$. X_i is the complete set of features of a sample, and $j = 1, \dots, m$, and $i = 1, \dots, M$, also $|x_j| = N_{bin}$. In this case m is the total number of bins (N/N_{bin}) of a sample and M is the total number of samples. Let F represent the desired feature set, where a bin is defined as a feature, and let z represent the number of features of F ($|F| = z$). Suppose that $H(\cdot)$ is the feature set evaluation function and that maximizing H produces better discriminating feature sets. Feature set redundancy is considered also, therefore, let $J(\cdot)$ be a function that determines redundancy and minimizing $J(\cdot)$ is desired. The feature set selection problem is finding a set of features $F \subseteq Y$ such that $|F| = z$ and

$$H(F) = \max_{Q \subseteq Y, |Q|=z} H(Q) \quad (3.1)$$

and simultaneously

$$J(F) = \min_{Q \subseteq Y, |Q|=z} J(Q). \quad (3.2)$$

Our desire is to determine F in a stochastic, non-greedy manner. NASAFS and NASAFS-IDF accomplishes this task.

NASAFS determines a feature set based on the set of features. Instead of determining the best features based on some evaluation criteria and placing them in a set, a set of features is chosen and the set is evaluated. This process may choose individual features that alone might not be the best class discriminators; however, when placed within a group of features, this group may outperform a feature set of individually great discriminating features. When correctly incorporated, the performance of the group method of feature selection can take advantage of synergies of features within the set, producing highly accurate class discrimination.

NASAFS consists of three distinct stages: *selection*, *evaluation*, and *candidacy determination*. The selection stage comprises of two different methods: an initial selection and an iterative selection. The evaluation stage is determined by the distributed spacing method and the heuristic. The heuristic determines the discrimination capability of the feature set and consists of dependence and distance measures (Eqn. 3.19). The candidacy determination stage is judged by the search algorithm, simulated annealing. These three stages are repeated until convergence.

3.1.1 NASAFS Methodology. Fig. 3.2 is a flow diagram that aids in the understanding of the NASAFS process and works as follows:

1. Compute the correlation matrix, r_X . Bin the data, $x_j \subseteq X_i$ where x_j is the j^{th} bin of the data sample X_i and $|x_j| = N_{bin}$ and $i = 1 \dots M$ where M is the total number of samples. Determine the cross-covariance threshold, $k_h = \min[\text{xcov}(x_{j,i}, x_{j,K})]$, where $x_{j,i}$ is the j^{th} bin of the i^{th} sample, and i and K are the same class and $i = 1 \dots M$, $j = 1 \dots m$ where m is the total number of bins of a sample, and K is defined as $\{K = 1 \dots M : K \neq i\}$ (Section 3.1.1.1).
2. (a) *Randomly* select a feature set F , $F \subset Y$ where $Y = \{x_1, x_2, x_3, \dots x_m\}$ and $|F| = z$ where z is defined by the user. (b) Ensure acceptable distribution across the signal domain, $\iota < \iota_u$, ι_u is the acceptable distribution (Section 3.1.1.2).
3. (a) Heuristic, h , evaluates the feature set (Eqn. 3.19), (b) Compute the return scalar value using the simulated annealing search (Section 3.1.1.3).
4. (a) Replace a feature in the feature set with a random pick of the remaining features. (b) Maintain the distributed spacing requirement. (c) Evaluate the new feature set with the heuristic (h), (Section 3.1.1.4).
5. Compare feature sets with SA (Section 3.1.1.4).
6. Repeat steps 4 and 5 until convergence (Section 3.1.1.4).

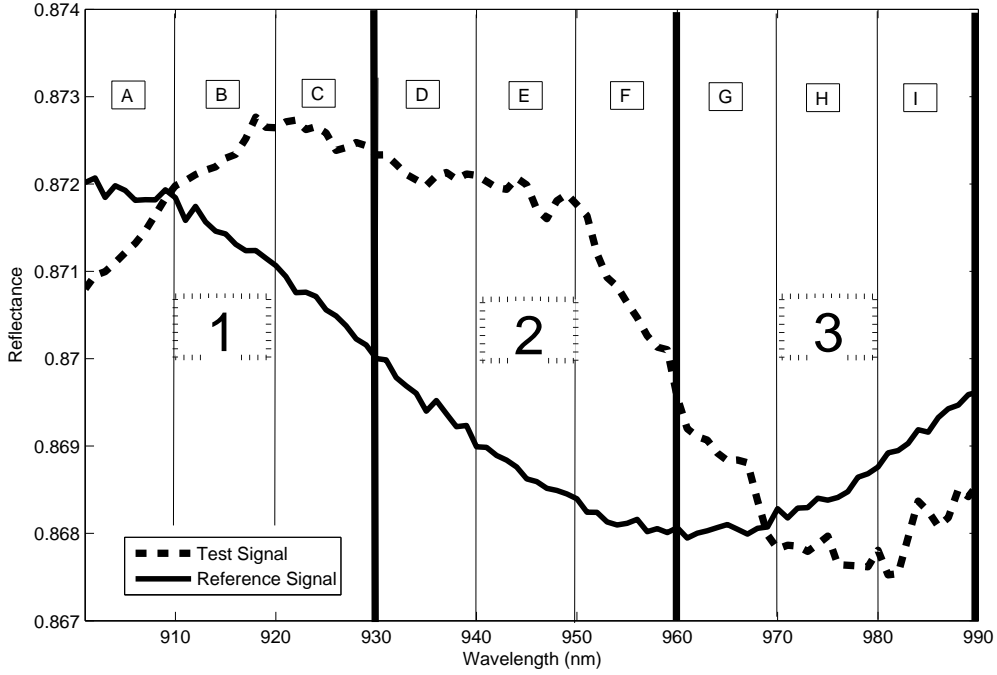


Figure 3.1: Example of a hyperspectral signal segment, divided into bins and sub-regions. The solid line is the reference signal and the dashed line is the test signal; 1...3 are the sub-regions and A...I are the bins. Both signals have a $1nm$ sampling interval and each bin is constructed of 10 spectral attributes.

3.1.1.1 Step 1: Training (Feature Selection Covariance Threshold).

NASAFS trains on the reference class samples by finding the lowest cross-covariance value, $(xcov(x, y))$, (shown in Eqn 3.4) of every combination of each corresponding bin of the reference class, Fig. 3.1 and 3.3. This cross-covariance threshold is determined as:

$$k_h = \min[xcov(x_{j,i}, x_{j,K})] \quad (3.3)$$

where $x_{j,i}$ is the j^{th} bin of the i^{th} sample, and i and K are the same class where K is defined as $\{K = 1...M : K \neq i\}$, and $i = 1...M$, $j = 1...m$ where m is the total number of bins of a sample. Where the bin is defined as, $x_j \subseteq X_i$ where x_j is the j^{th} bin of the data sample X_i and $|x_j| = N_{bin}$. Figs. 3.1 and 3.3 provide a visual

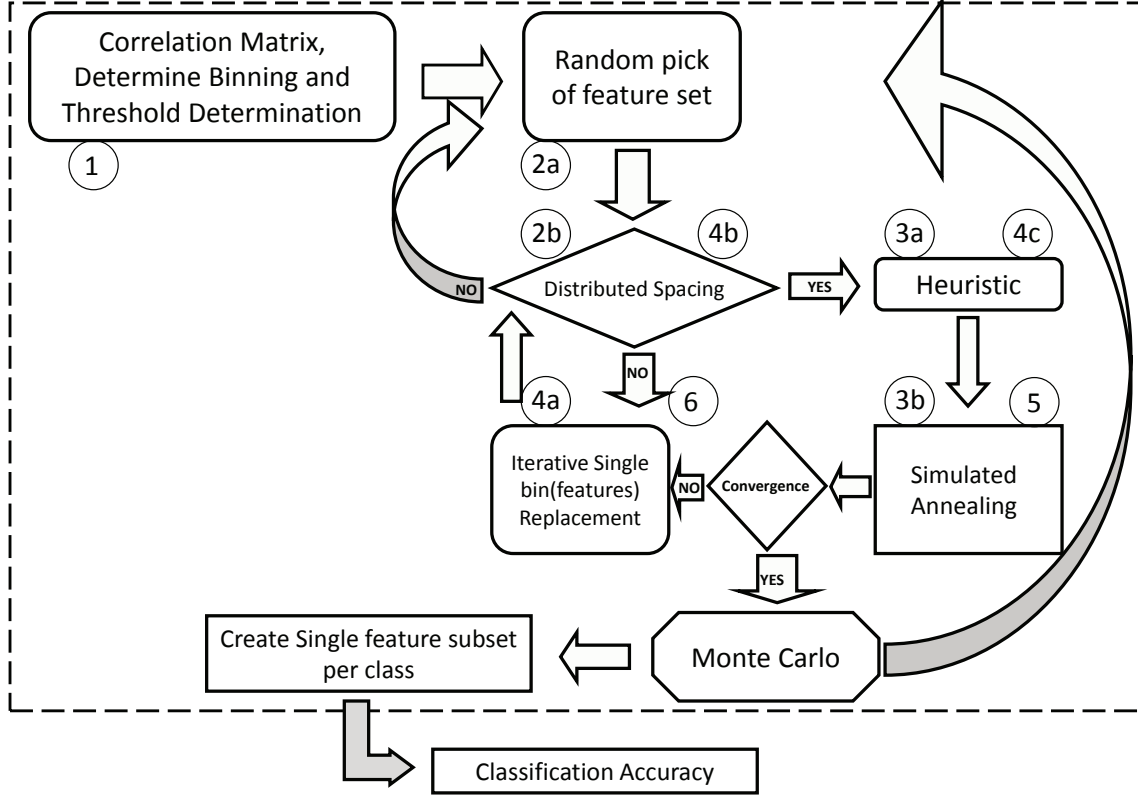


Figure 3.2: Diagram depicting the flow of feature selection method NASAFS (within the dashed box), and its connection to a classification method (outside the dashed box).

understanding of the *bin* concept. The cross-covariance is defined as:

$$\begin{aligned}
 xcov_{x,y} &= E[(x - \mu_x)(y - \mu_y)] \\
 &= \rho_{x,y} - \mu_x\mu_y
 \end{aligned}
 \tag{3.4}$$

where $E[\cdot]$ is the expectation, x and y are random variables, μ_x and μ_y are the means of the random variables, and ρ is the cross-correlation [52].

The process described is used for both the high-resolution and low-resolution data sets. However, the determination of bins is different for each resolution case. The high-resolution case uses non-overlapping (sequential) bins, as shown in Fig. 3.1. In the low-resolution case, the bins divide the signal by sliding the bin over, one attribute or dimension at a time, as shown in Fig. 3.3. The training function requires

at least two samples from the reference class to assess a covariance threshold k_h ; this threshold is then used in the heuristic, shown in Eqn. 3.19. The user must define the bin size, which is typically set to the bandwidth of the target collection system to be used for the detection task.

NASAFS uses the construct of features, which has a specific meaning as it applies to hyperspectral data or a correlated continuous domain; however, for this work, a feature is a grouping of consecutive dimensions of a sample. This grouping of consecutive dimensions are considered a *bin*; therefore, a feature set is a set of features where each feature is a bin and each bin can be two or more sequential attributes of the signal. The purpose of a bin is to divide the signal for processing by NASAFS (Fig. 3.1 and 3.3).

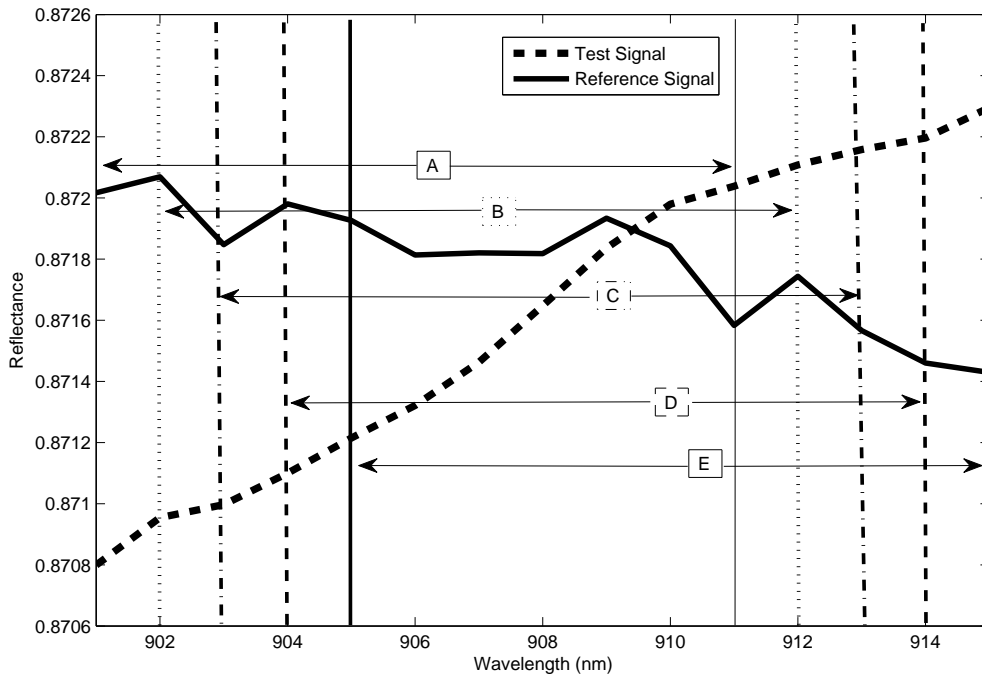


Figure 3.3: Example of a hyperspectral signal segment; divided into bins and sub-regions. The low-resolution data uses a sliding window technique instead of a sequential binning process. Each letter corresponds to a different bin; the solid line is the reference signal, and the dashed line is the test signal, and A...E are the bins. The example shown is for a $10nm$ bin size consisting of 10 spectral attributes per bin.

3.1.1.2 Step 2: Distributed Spacing. Distributed spacing is a technique used to divide a population into sub-regions. These sub-regions are then used to determine if the total population is distributed as per expectation. To adapt the concept of distributed spacing to feature selection, it is necessary to have an understanding of the problem this equation is intended to solve. The feature selection methods described in Section 2.2 have a tendency to select highly correlated features. The goal is to incorporate a measure of feature distribution that optimizes feature selection to prevent correlated feature sets. This approach will produce a better feature set that properly represents a class, resulting in increased discrimination accuracy, especially in the presence of noise.

The equation presented in the book by Coello Coello *et al.* is intended for use with Multi-Objective Evolutionary Algorithms (MOEA) [12]. The variables in the equation given in [12] have specific meanings that are tied to that domain; most are in regards to a genetic algorithm. The purpose of the distributed spacing equation is to determine how well points are distributed in relation to the optimum point distribution.

In its original context, distributed spacing provides a measure of how distributed the Pareto optimal solutions are across the MOEA non-dominated region [12]. A solution x^* is Pareto optimal if it establishes an equilibrium of sorts. This means there is no other x solution that can decrease a criterion without causing another criterion to increase [12]. In the description, x is a N -dimensional vector that has N decision variables and where x^* is the global minimum solution. The vectors that correspond to the Pareto optimal solutions are termed as non-dominated. For example, if two vectors are non-dominated, neither vector is better than the other; however, if vector x_1 is slightly better than vector x_2 , then x_1 dominates x_2 [85]. Any group of vectors that is not dominated by any other vector is termed non-dominated [85]. Therefore, optimal solutions of a multi-objective optimization problem are the non-dominated solutions.

The sub-regions, presented in the definition of the distributed spacing equation in [12], are considered as multiple populations that contain several vectors (genotypes/individuals) within each population. The distributed spacing equation measures how well the Pareto optimal solutions (non-dominated solutions) are distributed across these populations; a group of populations constitute a region. The equation given in the book by Coello Coello *et al.* is explained by Srinivas and Deb [12, 85], and is:

$$\iota = \sqrt{\sum_{i=1}^{q+1} \left(\frac{n_i - \bar{n}_i}{\sigma_i} \right)^2} \quad (3.5)$$

where q is the number of desired optimal points and $(q+1)$ is the dominated sub-region, n_i is the actual number of individuals in the i^{th} sub-region of the non-dominated region, \bar{n}_i is the expected number of individuals serving the i^{th} sub-region of the non-dominated region, and σ_i^2 is the variance of the individuals serving the i^{th} sub-region of the non-dominated region [85]. To illustrate this point, consider the following example. A region is equally divided into five sub-regions, meaning each region has an equal amount of physical space. Assuming the total population of the region consists of 100 individuals, then the expected number of individuals in each sub-region would be $\bar{n}_i = 20$. According to Srinivas and Deb [85], the variance is estimated as:

$$\sigma_i^2 = \bar{n}_i \left(1 - \frac{\bar{n}_i}{P} \right) \text{ for } i = 1, 2, \dots, q \quad (3.6)$$

where P is the total population size, in this case 100. Thus, the variance is $\sigma_i^2 = 20(1 - \frac{5}{100}) = 16$. Srinivas and Deb also show that the variance for the dominated region is the sum of the variances of the non-dominated regions ($\sigma_{q+1}^2 = \sum_{i=1}^q \sigma_i^2$); ideally, distribution points would not exist in the dominated sub-region. Therefore, the expected value of the dominated region is $E[\bar{n}_{q+1}] = 0$ [85]. Now, we determine the actual number of individuals in each sub-region. If the number of individuals that actually exist in each sub-region equals the expected number, the result is zero.

Therefore, if the solutions are distributed optimally, the result of Eqn. 3.5 is zero. As the distribution diverges from the optimal solution, the value obtained from Eqn. 3.5 increases.

The region of domination is purposefully omitted when translating the distributed spacing equation to the feature selection domain. This allows for the non-dominated region to be interpreted as the entire hyperspectral signal (e.g., in the 12 class textile data set, 901nm-2500nm). The signal is then divided into sub-regions. The sub-region division can be determined via a trade study for the type/class of signal being operated on; this will determine the best locations for divisions, based on some distinction of non-correlated regions. For example, the visible spectrum of a hyperspectral signal can be divided into different color regions to provide clarification. For our purpose, the *individuals*, as coined by Srinivas and Deb, are represented as the selected features of our data. Based on these changes, the distributed spacing equation (Eqn. 3.5) provides a good representation of distributed spacing for a feature set in the data domain.

The adaptation of the distributed spacing from [12, 85] is defined as [11]:

$$\iota = \sqrt{\sum_{i=1}^q \left(\frac{|F_{sub(i)}| - |\overline{F}_{sub(i)}|}{\sigma_i} \right)^2} \quad (3.7)$$

where q is the number of sub-regions (*Fig. 3.1*), $|F_{sub(i)}|$ is the *actual* number of selected feature points in the i^{th} sub-region, $|\overline{F}_{sub(i)}|$ is the expected number of feature points in the i^{th} sub-region (if sub-regions are unequal in bandwidth, a weighting must be applied), and σ is the standard deviation such that σ_i is the standard deviation of the i^{th} sub-region and is calculated as $\sigma_i = \sqrt{\sigma_i^2}$, and σ_i^2 is the variance of the i^{th} sub-region, and is calculated as:

$$\sigma_i^2 = |\overline{F}_{sub(i)}| \left(1 - \frac{|\overline{F}_{sub(i)}|}{N} \right) \text{ for } i = 1, 2, \dots, q \quad (3.8)$$

where N is the total number of attributes or dimensions of a sample. NASAFS determines the number and location of sub-regions (q) based on the structure of the correlation matrix of the entire data set. If the sub-regions are unequal in size, then a weighting ($w_{sub(i)} = \frac{n_{sub(i)}}{N}$) is applied to $|\overline{F}_{sub(i)}|$, where $n_{sub(i)}$ is the number of dimensions (bandwidth) of the i^{th} sub-region, and N is the total number of dimensions (bandwidth) of the signal. The best distributed case (ι) is determined prior to the execution of the feature selection process. This best case value for ι (Eqn. 3.7) is the situation where the desired number of features of a feature set is divided equally into each sub-region. This is based on the fraction of the expected number of features per sub-region divided by the total of the expected number of features for the domain (rounding up since a feature can not be split between sub-regions). This produces the lowest possible ι value for this specific data set. The optimal ι (lowest ι value) is then the baseline for all other values. To equate the appropriate distributed spacing number for all other results of Eqn. 3.7, the baseline value is subtracted from each value calculated.

A restriction is placed on bin size, in order to have the possibility of optimally distributed features. This restriction is to ensure that the bin size is small enough, based on the sub-region size, to allow for the appropriate number of features to exist in each sub-region. This restriction on the bin size (η) is calculated as:

$$\eta = \lceil w_{sub(s)} |F| \rceil, \quad (3.9)$$

where $w_{sub(s)}$ is the weighting factor of the smallest sub-region, $|F|$ is the total number of allowed features, and $\lceil \cdot \rceil$ is the ceiling operator. This value of η is then divided into the dimension/bandwidth of the smallest sub-region $n_{sub(s)}$, as shown:

$$\beta = \frac{n_{sub(s)}}{\eta} \quad (3.10)$$

where $n_{sub(s)}$ is the smallest sub-region. Therefore, in order to have an optimal distribution of features, the maximum bin size allowed is β .

An equation for $|\overline{F}_{sub(i)}|$ is:

$$|\overline{F}_{sub(i)}| = \frac{|F|}{q} w_{sub(i)} \quad (3.11)$$

where $|F|$ is the feature set size, q is the number of sub-regions, and $w_{sub(i)}$ is the weighting for each sub-region.

To explain the equations as they pertain to this type of problem using unequal sub-regions, consider the following example. Given a sample with 100 dimensions ($N = 100$), the signal is divided into five sub-regions, where one sub-region consists of 40 dimensions and the other four sub-regions have 15 dimensions each. For this example, we will set the number of features the algorithm will select to six; this is a user-defined variable. Therefore, since sub-regions 1-4 are identical (same number of dimensions), the expected number of features per region for sub-regions 1-4 is $|\overline{F}_{sub(i)}| = \left(\frac{6}{5}\right) \left(\frac{15}{100}\right) = 0.18$ (Eqn. 3.11), and the variance is $\sigma_i^2 = 0.18 \left(1 - \frac{0.18}{100}\right) = 0.179$ (Eqn. 3.8), which gives a standard deviation of $\sigma_i = 0.423$, where $i = 1..4$. Similarly, for $|\overline{F}_{sub(5)}| = \left(\frac{6}{5}\right) \left(\frac{40}{100}\right) = 0.48$ (Eqn. 3.11), the variance is $\sigma_5^2 = 0.48 \left(1 - \frac{0.48}{100}\right) = 0.478$ (Eqn. 3.8), and the standard deviation $\sigma_5 = 0.691$. The maximum bin size is determined as: $\eta = (0.15)(6) = 0.9$ (Eqn. 3.9); this number is rounded up so that $\eta = 1$. Therefore, $\beta = \frac{15}{1} = 15$ (Eqn. 3.10). A bin size of five is selected, to avoid exceeding the maximum bin size of 15; however, any number less than 15 would be adequate in this case. Next, the best case situation is calculated; for sub-regions 1-4, allot one feature $|F_{sub(i)}| = 1$ where $i = 1..4$, and for sub-region 5, allot two features ($|F_{sub(5)}| = 2$). This presents the best distributed case for this situation. Therefore, the optimal ι value, rounded to the hundredths place, is:

$$\begin{aligned} \iota &= \sqrt{\left(\frac{1 - 0.18}{0.423}\right)^2 (4) + \left(\frac{2 - 0.48}{0.691}\right)^2} \\ &= 4.46(100) = 446. \end{aligned} \quad (3.12)$$

The value of $\iota = 446$ and is now the baseline; it is subtracted from the actual *iota* values obtained for any other calculated distribution of features. For the worst distributed case, it is observed that sub-regions 1-4 have zero features assigned to them $|F_{sub(i)}| = 0$ where $i = 1..4$, and sub-region 5 contains all six allotted features, such that $|F_{sub(5)}| = 6$. Since $|\overline{F}_{sub(i)}|, \sigma_i^2, \sigma_i$ and q remains the same. The new observed numbers for $|F_{sub(i)}|$, rounded to the hundredths place, produce:

$$\begin{aligned} \iota &= \sqrt{\left(\frac{0 - 0.18}{0.423}\right)^2 (4) + \left(\frac{6 - 0.48}{0.691}\right)^2} \\ &= 8.03(100) = 803. \end{aligned} \tag{3.13}$$

The baseline (best case ι value, 446) is subtracted from the worst case ι value (803) to obtain an adjusted worst case value of 357. The adjusted worst case value is significantly larger than zero. This is due to the fact that it is a highly correlated distribution of the features. Now the distributed spacing values are bracketed for this scenario, which determines the degree of optimal distribution of each feature set calculated.

In order to incorporate the distributed spacing equation, and to ensure the possibility of non-correlated selected features, the maximum number of features of a feature set is defined as $\sum_i^q |\widehat{F}_i|$, where q is the number of sub-regions and $|\widehat{F}_i|$ is the number of allowed features per sub-region. However, the variable $|\widehat{F}_i|$ is not the maximum number of features that a sub-region can hold; it is the maximum allowed number of features in each sub-region to ensure optimal distribution of features, where optimality is defined by minimizing ι (Eqn 3.7). The value of $|\widehat{F}_i|$ is calculated as $|\widehat{F}_i| = \lceil w_{sub(i)} N \rceil$ where $\lceil \cdot \rceil$ is the ceiling function, $w_{sub(i)}$ is the weighting of each sub-region, and N is the total number of dimensions of the data. The maximum number of features that a sub-region can hold is dependent on the bin size, which is determined by the user.

NASAFS ensures less correlated (non-redundant) feature sets by selecting features based on a specific distribution space across the spectrum. The key to this

process is determining how to divide the signal domain into appropriate sub-regions. If the correlation statistic is used, q (shown in Eqn. 3.7) is required to be the number of signal sub-regions, as determined by the correlation matrix of the entire data set.

Fig. 3.4 shows all the pairwise correlations of a set of hyperspectral data. The matrix is positive semi-definite about the diagonal [52], and its structure produces sub-regions of correlation. This structure is used by our method to produce the sub-regions for our distributed spacing ι function. The correlation matrix of the data, as shown in Fig.3.4, is calculated and the sub-regions are located automatically by NASAFS. The process of locating the sub-regions of the correlation matrix is dependent on preset variables, which can be adjusted for each data set. One of these variables defines the correlation value threshold that determines one sub-region from the next. The other two variables define a minimum width that must be surpassed in order to declare the next sub-region. For example, Fig. 3.4 illustrates the correlation matrix of the 12 class textile data set. The threshold is set to 0.85 and the row variable is 50, while the column variable is 100. For the 12 class textile data case, the method finds the index of the row in column 1 that is less than the threshold; it then places a marker at that location. It proceeds to row 2 column 2, and finds the index of the row in column 2 that is less than the threshold, and places a marker there. This process proceeds down the diagonal for the entire matrix. Fig. 3.5 is a illustration of the markers and their locations, based on the 12 class textile data set correlation matrix. After all the markers are placed, the process then finds the column where the sequential marker is separated by 50 rows. From that location, the process proceeds along that row to the point 100 columns over from its original location. If it is determined that, from the original location to the 100 subsequent locations, there are no other markers with a row index less than the original marker index (assuming row 0, column 0 is in the upper left corner), then a sub-region is established at the location of the original marker. Then the process moves 101 columns from the newly determined sub-region and repeats. If it is determined that there is a marker with a row index less than the original marker index (within the 100 columns), the process starts over at the next

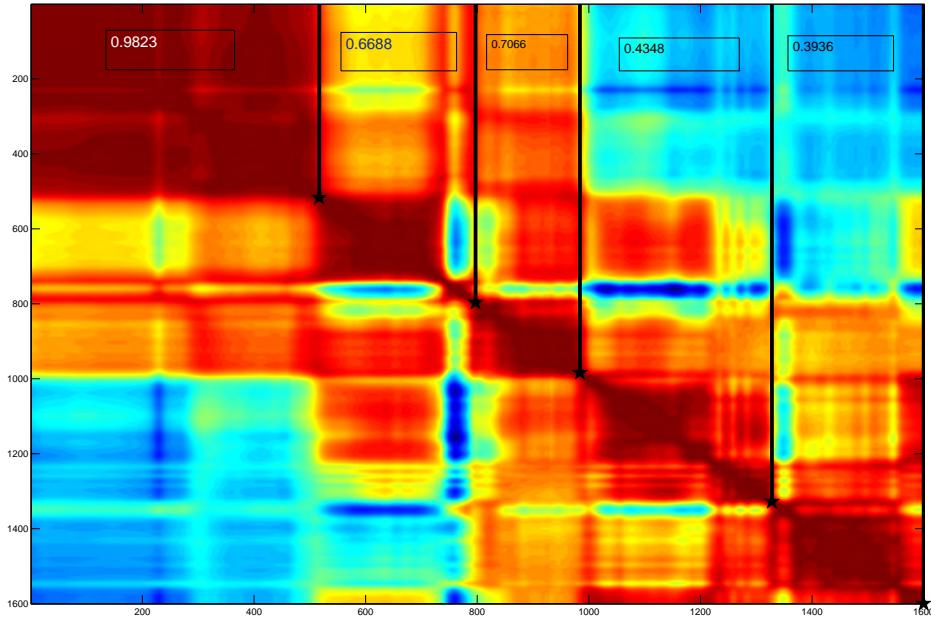


Figure 3.4: Correlation matrix of the 12 class textile data set. The lines denote the different sub-regions and the average correlation value of that region is shown in the box. The row and column variables for the sub-region locator method are 50 and 100, respectively.

marker location from the original marker. Fig. 3.6 shows the sub-regions of the 12 class textile data set, if the column and row variables are 10 and 25, respectively.

The best case (ι) is calculated, where best case ι provides the least-correlated case that is based on the number of desired features in the feature set and the number of sub-regions of the spectral domain (e.g., if the feature set is to contain six features, then for the case in Fig. 3.1, there would be two features per sub-region). As previously stated, the optimal (ι) is used as a baseline when determining the actual correlation of the selected feature set. Ideally, a value of zero for the distributed spacing (ι) equation indicates a perfect distribution across the domain. However, the best case distributed spacing value might not be zero, depending on the divisions created by the sub-regions and the number of features designated for the feature set. In order to obtain an indication of optimality when each (ι) value is calculated, the optimal

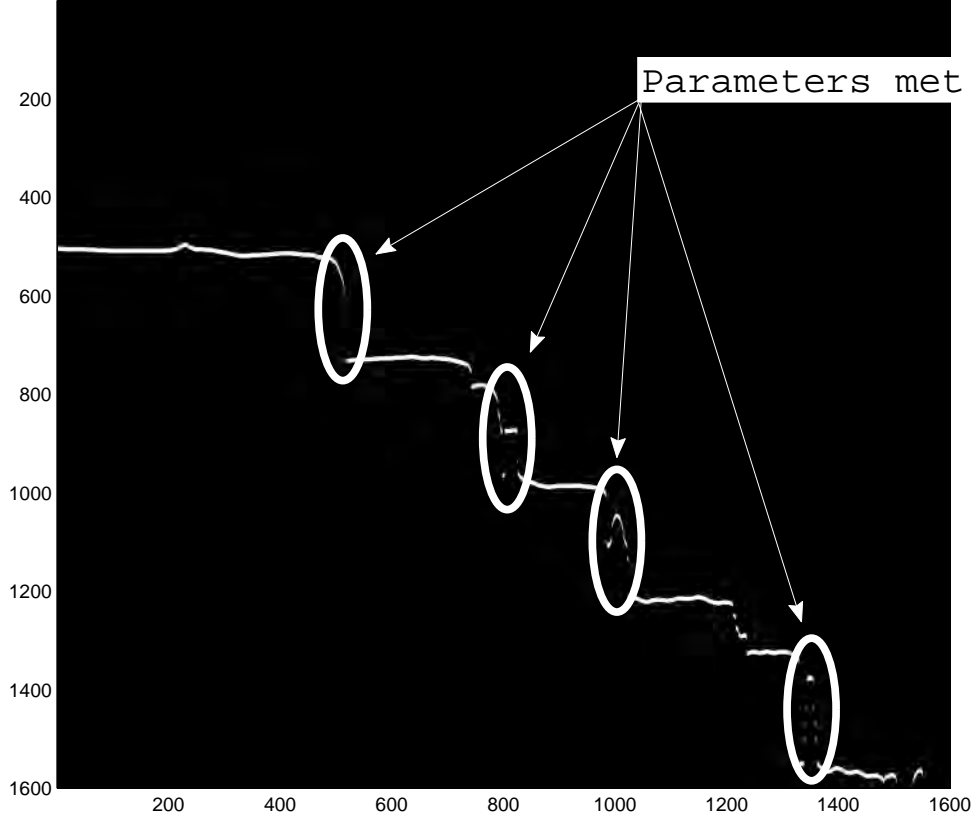


Figure 3.5: This figure illustrates the placement of the markers for determining the sub-regions of the correlation matrix. The row and column variables for the sub-region locator method are 50 and 100, respectively.

distributed case and the worst distributed case are determined first. The best case (ι) value is then subtracted from each calculated value of (ι) to determine the degree of optimality produced by each feature set. The best and worst case values provide the upper and lower bounds; these are useful in determining an acceptable percentage of distribution. The actual (ι) calculation can now be evaluated as a percentage, allowing for the user to define a desired percentage of distributed spacing for a feature set.

The best case (ι) is determined by allotting one feature per sub-region, starting with the sub-region that has the lowest average correlation value. The process continues until the number of desired features is exhausted. If the number of desired

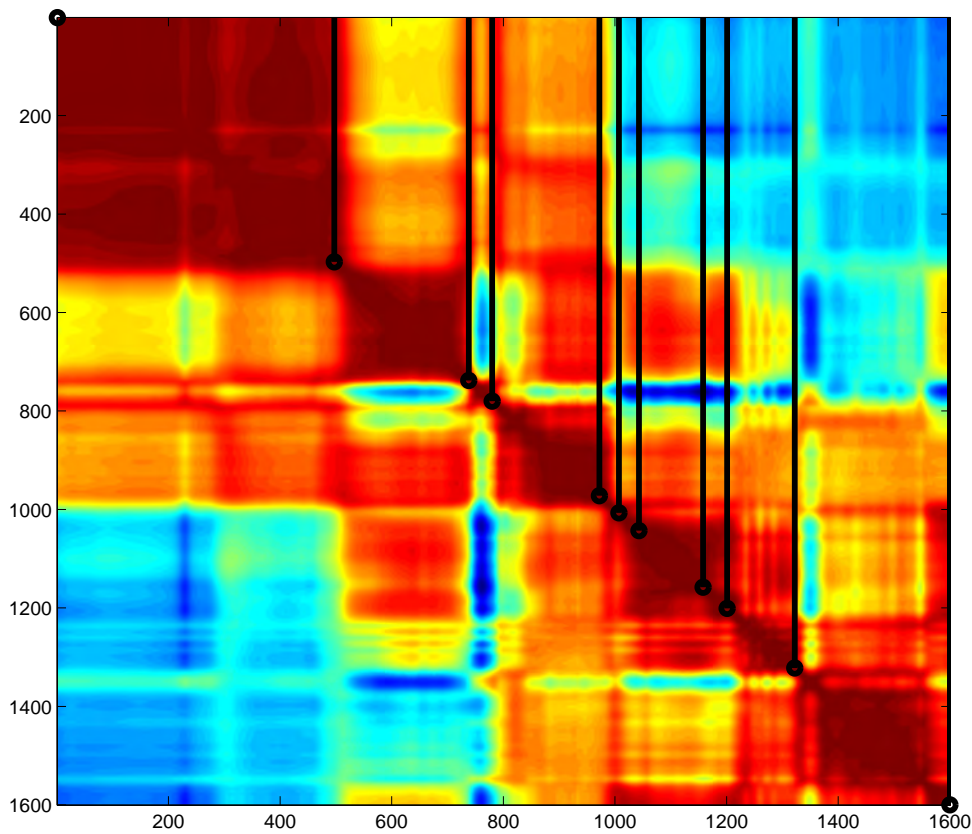


Figure 3.6: Correlation matrix of the 12 class textile data set. The lines denote the different sub-regions, where the row and column variables for the sub-region locator method are 10 and 25, respectively.

features is greater than the number of sub-regions, the process continues allotting features to sub-regions, beginning with the sub-region that has the lowest correlation value; in this instance, features will be allotted to a sub-region until that sub-region reaches its expected number of features ($|\overline{F}_{sub(i)}|$). The worst case (ι) is calculated by allotting all of the number of desired features to the sub-region that has the worst (highest) average correlation value.

The expected number of features in each sub-region ($|\overline{F}_{sub(i)}|$) is determined by the average correlation value and the spectral bandwidth of the i^{th} sub-region; this is

computed as:

$$|\bar{F}_{sub(i)}| = \frac{\left(|F| * \left(\frac{n_{sub(i)}}{N}\right) + |F| * \left(\frac{1-\bar{r}_{sub(i)}}{\sum_j 1-\bar{r}_{sub(j)}}\right)\right)}{2} \quad (3.14)$$

where $|F|$ is the feature set size, \sum_j is the sum over all sub-regions, $n_{sub(i)}$ is the spectral bandwidth of region i , N is the total spectral bandwidth of the signal, and since \bar{r} is the average correlation coefficient, then $\bar{r}_{sub(i)}$ is the average correlation coefficient of each region. In Eqn. 3.14, $\frac{n_{sub(i)}}{N}$ is the fraction of bandwidth of sub-region i and $\frac{1-\bar{r}_{sub(i)}}{\sum 1-\bar{r}_{sub(i)}}$ is the fraction of the correlation value of sub-region i .

If a feature set meets the acceptable optimal spacing, it is allowed to proceed to the heuristic. If not, that feature set is discarded; a new feature set is then selected at random and the process is repeated.

3.1.1.3 Step 3: The Heuristic. The heuristic has a two-fold approach to determine a feature set's *goodness*. The first part encompasses dependence measures, and the second uses distance measures. Combining these two techniques is the first step to optimizing the evaluation function. The distance measure determines a distance between an in-class feature set and the respective out-of-class feature set. If the distance exceeds a previously established threshold, a high value is returned for that feature set. The dependence measure uses statistics to stratify the selected feature set. The value returned for that feature set is dependent on the correlation value between the classes. The dependence measure component of the heuristic consists of comparing the cross-covariance of the two feature sets (the reference class and the target class). The cross-covariance for the heuristic is specifically defined as:

$$xcov_{x_{ref}, x_{tgt}} = E[(x_{ref} - \mu_{x_{ref}})(x_{tgt} - \mu_{x_{tgt}})] \quad (3.15)$$

where $E[\cdot]$ is the expectation, x_{ref} is the reference feature set, x_{tgt} is the target feature set, $\mu_{x_{ref}}$ and $\mu_{x_{tgt}}$ are the means of the reference and target feature sets, respectively. If the cross-covariance is one, then the feature sets are highly correlated and the

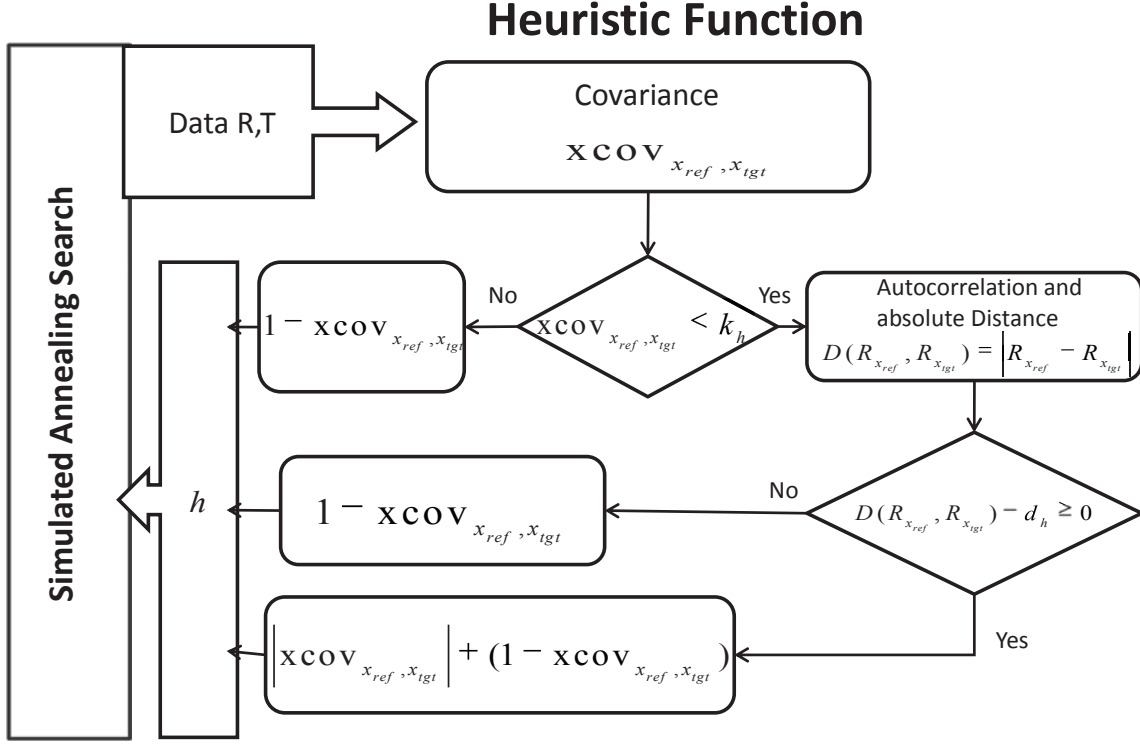


Figure 3.7: Flow chart for the heuristic function of NASAFS.

two feature sets are less correlated as the value of the cross-covariance deviates from one. The cross-covariance is preferable to the cross-correlation in terms of providing a better measure of comparison between the two spectral measurements. It can be shown from Eqn. 3.15 that the cross-covariance is equal to the cross-correlation minus their means. This neutralizes any bias that can occur from differing signal strengths.

The heuristic returns values based on a sequential set of calculations. A flowchart of the heuristic is shown in Fig. 3.7. The cross-covariance $xcov_{x_{ref}, x_{tgt}}$ (Eqn 3.15) is calculated using a feature set of the reference class that corresponds to the same feature set of the target class (Fig. 3.1). The cross-covariance is calculated using the same process as is used in the training function portion of the algorithm (see Section 3.1.1.1). However, during training, each bin is evaluated individually. In the heuristic, the cross-covariance function evaluates the feature set (where each feature, of the set, is a bin), not each individual feature. The cross-covariance of the feature set is accomplished for the target samples versus the reference samples. This value is

compared to the covariance threshold k_h . If $xcov_{x_{ref},x_{tgt}} \geq k_h$, then $(1 - xcov_{x_{ref},x_{tgt}})$ is returned per Eqn. 3.19. This means, if the covariance indicates the two feature sets are highly correlated, then the returned value corresponding to that set of features is low. If the value of the covariance is less than k_h , then those two feature sets continue to the autocorrelation section of the heuristic. The autocorrelation R of a random variable X , (R_X), is defined as:

$$\begin{aligned} R_X &= E[X(t_1)X(t_2)] \\ &= \sum_k \sum_r x_k y_r p_{X(t_1),X(t_2)}(x_k, y_r) \end{aligned} \quad (3.16)$$

where $E[\cdot]$ is the expectation, $X(t_1)$ and $X(t_2)$ are random variables that are time shifted, $p_{X(t_1),X(t_2)}(x_k, y_r)$ is the multivariate probability mass function of $X(t)$, and t_i is the time shift. The autocorrelation determines similarity between observations of the same signal that are shifted in time [52]. It is useful in this application because it acts as a kernel. If there is a difference between the two signals, the distance between them is exaggerated. The absolute distance, D , between the autocorrelation of the target feature set value and the autocorrelation of the reference feature set value, $D(R_{x_{ref}}, R_{x_{tgt}})$, is computed as:

$$D(R_{x_{ref}}, R_{x_{tgt}}) = |R_{x_{ref}} - R_{x_{tgt}}| \quad (3.17)$$

where $|\cdot|$ is the absolute value. If the distance $D(R_{x_{ref}}, R_{x_{tgt}})$ is less than a threshold d_h , $(1 - xcov_{x_{ref},x_{tgt}})$ is returned to the search algorithm. If it is greater than the threshold, then $|xcov_{x_{ref},x_{tgt}}| + (1 - xcov_{x_{ref},x_{tgt}})$ is returned to the search algorithm. The threshold (d_h) is initially set by the user; however, if d_h is exceeded, the threshold is updated; this forces the next set of features to be a better solution than the previous set, as in:

$$d_h = d_h + \frac{|d_h - D(R_{x_{ref}}, R_{x_{tgt}})|}{2}. \quad (3.18)$$

The heuristic h is expressed as:

$$h = \begin{cases} 1 - xcov_{x_{ref}, x_{tgt}} & \text{if } xcov_{x_{ref}, x_{tgt}} \geq k_h, \\ |xcov_{x_{ref}, x_{tgt}}| + (1 - xcov_{x_{ref}, x_{tgt}}) & \text{if } xcov_{x_{ref}, x_{tgt}} < k_h \text{ and } D(R_{x_{ref}}, R_{x_{tgt}}) - d_h \geq 0, \\ 1 - xcov_{x_{ref}, x_{tgt}} & \text{if } xcov_{x_{ref}, x_{tgt}} < k_h \text{ and } D(R_{x_{ref}}, R_{x_{tgt}}) - d_h < 0. \end{cases} \quad (3.19)$$

The following is an example of the heuristic. If the cross-covariance value of the target feature set to the reference feature set is 0.54, and the covariance threshold is set to 0.85, then the feature sets are allowed to pass to the distance measure portion of the heuristic. If the value of the distance measure for the two feature sets is 0.4, and the distance threshold is $d_h = 0.1$, then for this feature set, a value of 1 is returned to the feature selection function; the new distance measure threshold is then updated to $d_h = 0.1 + \frac{|0.1-0.4|}{2} = 0.25$.

3.1.1.4 Steps 4-6: Feature Set Updates. This step begins the three stage search process. While the use of each stage is non-repetitive initially, it is repeated until convergence. The initialization process starts with NASAFS *selecting* a specific number of random features as the starting feature set (where each feature, of the set, is a bin). The first part of the *evaluation* process is determining if the feature set meets the accepted optimal spacing (Eqn. 3.7 and Section 3.1.1.2). The second part of the *evaluation* process is accomplished by the heuristic (Fig. 3.7 and Section 3.1.1.3). The third stage, *candidacy determination*, occurs when the heuristic value is returned to the simulated annealing function. Once all three stages have been

accomplished and an initial feature set has been successfully obtained, the three stage process (*selection, evaluation, and candidacy determination*) repeats. One feature is replaced by a randomly selected feature. This new set of features is evaluated for desired distributed spacing. Once this spacing is achieved, the feature set is evaluated and given a value by the heuristic; this value is then returned to the simulated annealing process for candidacy determination (Eqn. 3.19). The process of replacing features of the feature set is repeated until convergence. Once convergence is obtained, the current set of features becomes the solution.

Simulated annealing determines the retention of a feature set based on the value returned by the heuristic, the higher the value, the greater the probability of retention. However, it is possible that an inferior feature set might be kept in hopes of achieving a better solution on the next iteration. This probability is based on the evaluation of an exponential function $exp^{\frac{err}{T}}$, where err is the new feature set's heuristic value minus the previous feature set's heuristic value, and T is the time decayed *temperature* (based on the number of iterations of the search). The random chance is determined by generating a random value (Y_{rand}) between zero and one; if this value is less than or equal to the exponential value, the feature set is kept, as shown:

$$F = \begin{cases} F_{new} & \text{if } Y_{rand} \leq e^{err/T}, \\ F_{previous} & \text{if } Y_{rand} > e^{err/T} \end{cases} \quad (3.20)$$

where F_{new} is the new feature set, $F_{previous}$ is the previous feature set, and F becomes the current feature set. The time decayed *temperature* (T) is decreased by a factor of 0.9 for each time a new feature set is evaluated. The simulated annealing pseudo-code is shown in Algorithm 1 [74].

3.1.2 Final Feature Selection Process. This is a process of selecting features over several Monte Carlo simulations and across multiple class pairwise feature selections. It uses solution frequency distributions (histograms) to determine the features common to all pairwise histograms for each reference class. This methodology

Algorithm 1: Simulated Annealing pseudo-code [74]

```
function SA(problem,schedule) returns a solution state
input  : problem, a problem
         schedule, a mapping from time to “temperature”
local variables : current, a node
                   next, a node
                   T, a “temperature” controlling the probability of downward steps

current  $\leftarrow$  Make – Node(Initial-State[problem])
for t  $\leftarrow$  1 to inf do
  | T  $\leftarrow$  schedule(t)
  | if T = 0 then
  |   | return current
  |   next  $\leftarrow$  a randomly selected successor of current
  |    $\Delta E \leftarrow$  Value(next) - Value(current)
  |   if  $\Delta E > 0$  then
  |     | current  $\leftarrow$  next
  |   else
  |     | current  $\leftarrow$  next only with probability  $e^{\Delta E/T}$ 
```

is based on the correlation matrix of the data and the distributed spacing techniques discussed earlier.

First, this requires two user-defined variables: H_{hit} and A_{val} . The H_{hit} is constrained from zero to one and is the ratio of the number of times a feature is selected (across all histograms for that reference class) to the total number of histograms for that reference class. For example, if there are ten histograms for one reference class, then an H_{hit} value of 0.8 could be used. This means that if a specific feature is selected in eight (or more) out of ten histograms, it is possible it will be a good discriminating feature for that reference class. Ideally, an H_{hit} value of 1 would be used; however, it is not likely that a specific feature will be selected in all histograms. Example histograms can be seen in Fig. 3.8. The A_{val} is represented as a percentage that is based on the largest specific class ordinate value out of all the features in all the histograms. It represents the minimum acceptable number of times that a feature is selected as a member of a feature set for a specific reference class. It can also be defined as

the number of times that a feature was part of the feature set that NASAFS chose for each of the Monte Carlo simulations. A_{val} ensures that features available to be selected are not a chance select.

In order to ensure the distributed spacing of the feature set selected, This technique uses the same sub-regions as NASAFS. The feature in each sub-region that meets the H_{hit} criteria and has the best value in that sub-region is chosen. Once it has determined this for all the sub-regions, the number of features for the feature set is selected based on the features with the greatest value. The features are placed until the desired number of features for the set is obtained. If the number of features needed for the feature set is greater than the number of sub-regions by one feature, then it picks the extra feature based on the next highest valued attribute ($T_{next}(high)$) across all sub-regions, where T_{next} is the *next attribute*. If more than one feature is required to finish out the feature set, it then picks the largest ordinate valued attribute ($T_{next}(high)$) and compares it to the largest ordinate valued attributes of the other sub-regions. The attributes with the lowest correlation coefficients are selected. These two attributes are then added to the feature set. This process is repeated until the feature set is satisfied or until none of the remaining attributes meet the user-defined H_{hit} and A_{val} criteria. In some cases, the feature set may not be completely filled. The NASAFS pseudo-code is shown in Algorithm 2.

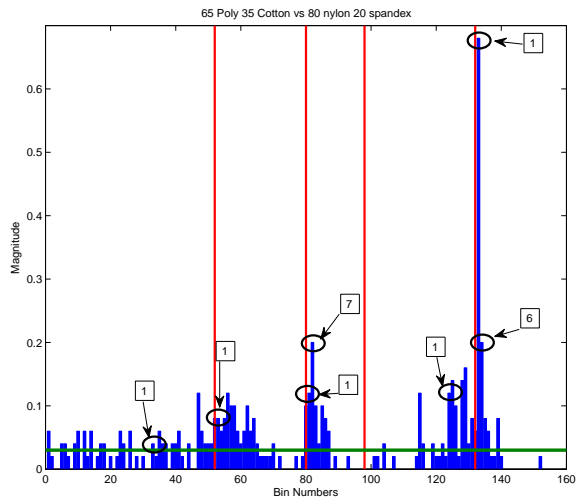
To further understand this process, consider the following example; for this instance, refer to Fig. 3.8, using an H_{hit} of 0.75 and an A_{val} of 0.05, and a feature set size of seven. This process determines all the features that are hit at least three out of the four times and that have a value greater than 0.05; it does this for each region of all four of the histograms. The sum of the values of the features that pass the H_{hit} and A_{val} criteria is used to determine the overall feature value. This overall feature value is used to select the feature with the largest value for that region; this is noted in Fig. 3.8 by the circle labeled 1 in each region. Since there are only five regions and seven features for the feature set, there are two more features to pick. The feature with the largest overall value that passed the H_{hit} and A_{val} criteria and was

Algorithm 2: Non-correlated Aided Simulated Annealing Feature Selection (NASAFS) pseudo-code

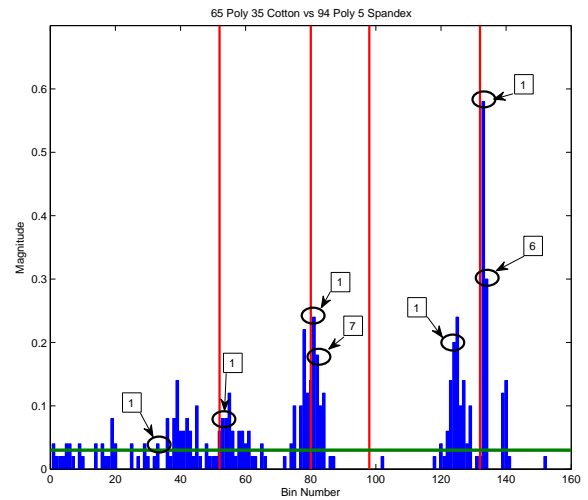
Input: Data set of samples with labels and user defined variables(udv)
Output: Feature set for a class versus all other classes
CMAT \leftarrow matrix of pairwise data correlation coefficients
// Bin data, see Fig. 3.1 and 3.3
DataBinned \leftarrow bin data based on BinSize(udv)
SubRegions \leftarrow Defined from CMAT based on regions of significant correlation, see Section 3.1.1.2 and Fig. 3.4
while NumClasses \neq Number of target classes **do**

INITIALIZATION
begin
// Compute cross-covariance threshold
for $j = 1 \dots$ number of bins **do**
 for $i = 1 \dots$ number of reference samples **do**
 $k_h[j, i] \leftarrow xcov(x_{j,i}, x_{j,K})$
 $k_{h(min)} \leftarrow$ minimum of k_h
// Monte Carlo Simulation
while Monte < NumRuns(udv) **do**
 // Select initial feature set F
Iota \leftarrow Value of worse case distributed spacing, see Section 3.1.1.2
// Iota must meet the distribution acceptability
while Iota > DistAcceptable(udv) **do**
 F \leftarrow Randomly select NumFeatures(udv) new features
 Iota \leftarrow Value of F *// Based on distributed spacing Eqn. 3.7*
Convergence \leftarrow No *// Initialized to 'no'*
while Convergence = no **do**
 EVALUATION
 begin
 // heuristic evaluates feature set, determine h
 $D_h \leftarrow 0.002$ *// Initialized distance threshold*
if ($xcov(x_{ref}, x_{tgt}) \geq k_{h(min)}$) or ($xcov(x_{ref}, x_{tgt}) < k_{h(min)}$ and $Dist(R_{x_{ref}}, R_{x_{tgt}}) < D_h$) **then**
 $h \leftarrow 1 - xcov(x_{ref}, x_{tgt})$
if ($xcov(x_{ref}, x_{tgt}) < k_{h(min)}$) and $Dist(R_{x_{ref}}, R_{x_{tgt}}) \geq D_h$ **then**
 $h \leftarrow |xcov(x_{ref}, x_{tgt})| + 1 - xcov(x_{ref}, x_{tgt})$
 $D_h \leftarrow$ increased
 // R is the aurocorrelation and Dist is the Manhattan distance
 CANDIDACY DETERMINATION
 begin
 SA $\leftarrow h, T$ *// see SA pseudo code Alg. 1*
 T $\leftarrow T \times$ schedule *// decrements T via schedule*
 SELECTION
 begin
 // Select a new feature set
Iota \leftarrow Value of worse case distributed spacing, see Section 3.1.1.2
while Iota > DistAcceptable(udv) **do**
 F \leftarrow Replace one feature with a randomly picked feature
 Iota \leftarrow Value of F *// Based on distributed spacing Eqn. 3.7*
 $F_{Histo(ref_i)} \leftarrow$ current F *// Histogram of feature sets for each reference class*
 NumRuns \leftarrow decrement NumRuns

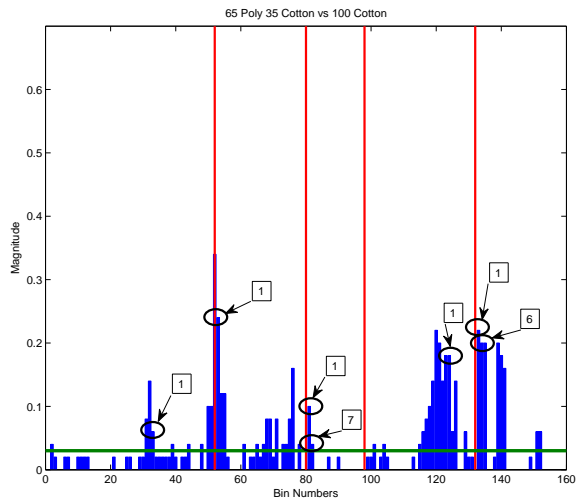
MasterHisto \leftarrow combination of all histograms ($F_{Histo(ref_i)}$) *// based on $H_{hit}(udv)$ and $A_{val}(udv)$*
 $F_{master} \leftarrow$ **the** top feature of each sub-region of MasterHisto until NumFeatures
while $|F_{master}| \neq$ NumFeatures **do**
 if NumFeatures = $|F_{master}| + 1$ **then**
 $F_{master} \leftarrow F_{master} +$ next highest feature over all sub-regions
 else
 $F_{master} \leftarrow F_{master} +$ next highest feature over all sub-regions with lowest correlation coefficient to last feature added to F_{master}



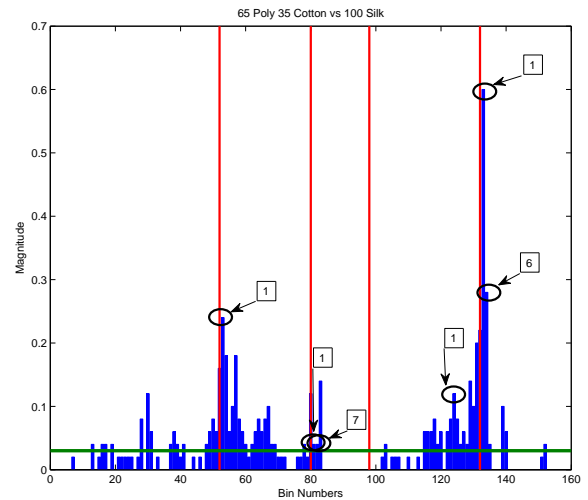
(a)



(b)



(c)



(d)

Figure 3.8: Histogram of selected features for: (a) 65% Polyester / 35% Cotton vs 80% Nylon / 20% Spandex, (b) 65% Polyester / 35% Cotton vs 94% Polyester / 6% Spandex, (c) 65% Polyester / 35% Cotton vs 100% Cotton, and (d) 65% Polyester / 35% Cotton vs 100% Silk. The red lines are the divisions of the sub-regions. The green line is the 0.05 (A_{val}) criteria line. The sub-regions labeled A - E have an average correlation coefficient of 0.9823, 0.6688, 0.7066, 0.4348, and 0.3936 respectively. The feature set selection process is indicated by the labeled circles for a feature set size of seven.

not picked on the first round is selected as the 6th feature; this is denoted in Fig. 3.8 by the circle labeled number **6**. Feature number 7 is determined by evaluating the correlation coefficient of the feature with the highest overall regional value, except the region with the 6th feature. The feature with the lowest correlation coefficient as compared to that of feature 6 is selected; this is shown in Fig. 3.8 as the circle labeled **7**. Note that for the first region of the histogram “65% Polyester 35% Cotton vs 100% Silk” (Fig. 3.8(d)), none of the features meet the H_{hit} criteria; therefore, the labeled **1** circle is omitted to emphasize this fact.

3.2 Non-correlated Aided Simulated Annealing Feature Selection - Integrated Distribution Function Overview

Our second version of NASAFS incorporates several improvements that reduces computation cost, increases accuracy and is more simplistic in design. The main contribution of the second version is the integration of the distributed spacing technique in the heuristic. NASAFS uses the distributed spacing method as a cross-check, where the second version uses the distributed spacing value as part of the value used by SA to determine feature set goodness. This integration requires several changes to the distributed spacing equation as well as the elimination of the correlation matrix and sub-regions. These improvements also provide the determination of one-versus-all feature set with fewer computations and only one histogram being created. Due to these improvements, the second version is called: Non-correlated Aided Simulated Annealing Feature Selection - Integrated Distribution Function (NASAFS-IDF).

While NASAFS and NASAFS-IDF are based on the same concepts, there are a some minor changes and two significant changes from NASAFS to NASAFS-IDF. One of the more important minor changes is based on the processing location of the one-versus-all approach. Both versions build a histogram of feature sets using Monte Carlo runs. However, NASAFS-IDF uses a one-versus-all approach that occurs in the feature selection stage, whereas NASAFS is a pairwise process until the final feature selection stage. This improvement eliminating the need to build pairwise histograms

of feature sets as in NASAFS. The other more important minor change is the need to determine the correlation matrix of the data. NASAFS-IDF also eliminates both the need for the correlation matrix and the assignment of sub-regions that NASAFS uses to guide the distributed spacing function. The first significant change in NASAFS-IDF is that the distributed spacing function is incorporated into the heuristic vector, which is optimized by the simulated annealing search method [4]. The second significant change is the new adaption of the distributed spacing function to accommodate this new paradigm. These improvements reduce computational costs and algorithm complexity; this increases both robustness and accuracy, while maintaining a low correlated feature set. The three stage process of selection, evaluation, and candidacy determination, as used in NASAFS, is also incorporated into NASAFS-IDF. However, the approach is more streamlined due to the improvements mentioned above.

3.2.1 NASAFS-IDF Methodology . NASAFS-IDF, shown in Fig. 3.9, works as follows:

1. Bin the data, $x_j \subseteq X_i$ where x_j is the j^{th} bin of the data sample X_i and $|x_j| = N_{bin}$ and $i = 1..M$, where M is the total number of samples. Determine the cross-covariance threshold, $k_h = \min[\text{xcov}(x_{j,i}, x_{j,K})]$, where $x_{j,i}$ is the j^{th} bin of the i^{th} sample, and i and K are the same class and $i = 1..M$, $j = 1..m$ where m is the total number of bins of a sample, and K is defined as $\{K = 1..M : K \neq i\}$ (Section 3.2.1.1).
2. *Randomly* select a feature set, $F \subset Y$ where $Y = \{x_1, x_2, x_3, \dots, x_m\}$ and $|F| = z$ where z is defined by the user.
3. (a) Heuristic, h_N , evaluates the feature set (Eqn. 3.25, Section 3.2.1.2). (b) Compute the return scalar value using the simulated annealing search (Section 3.2.1.3).
4. Replace a feature in the feature set with a random pick of the remaining features; evaluate the result with the heuristic h_N (Section 3.2.1.4).
5. Repeat steps 3 and 4 until convergence (Section 3.2.1.4).

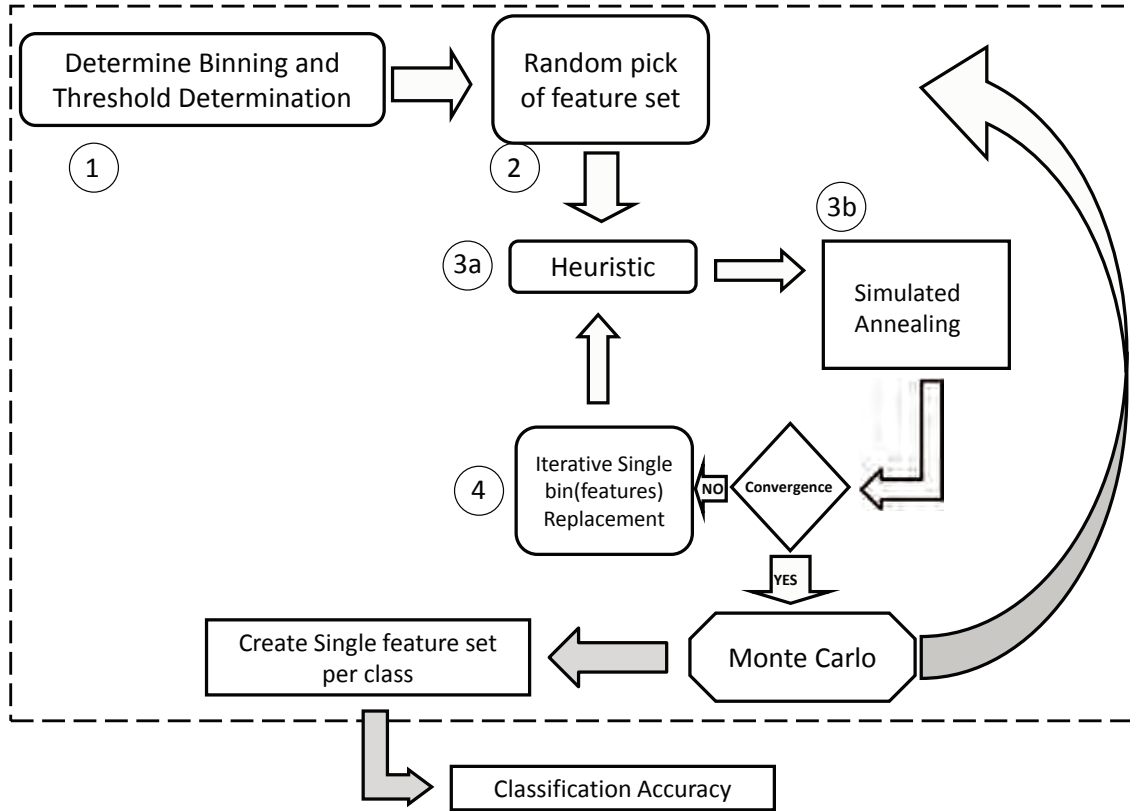


Figure 3.9: Diagram depicting the flow of feature selection method NASAFS-IDF (within dashed line), and its connection to a classification method (outside dashed line).

3.2.1.1 Step 1: Training (Feature Selection Covariance Threshold).

NASAFS-IDF trains on the data in the same manner as NASAFS (Section 3.1.1.1); however, where NASAFS employs sub-regions, NASAFS-IDF does not.

3.2.1.2 Step 2: Distributed Spacing.

The functionality of the distributed spacing technique is revised for NASAFS-IDF. In NASAFS, the distributed spacing function operated off of sub-regions of the data domain; in NASAFS-IDF, the distributed spacing operates without the constraints of the sub-regions across the data domain. The distributed spacing function of NASAFS-IDF calculates the degree of distribution across the entire domain (no sub-region constraints), allowing for a more dynamic distribution of features than is possible with NASAFS. This version of the distributed spacing technique closely resembles the original purpose of the technique

used in multi-objective optimization problems [12]. Instead of determining the distribution of Pareto optimal solutions across the non-dominated region, NASAFS-IDF determines the distribution of the features of a feature set across the data domain.

The distributed spacing equation (ι) for NASAFS-IDF is defined as:

$$\iota = \sqrt{\sum_{i=1}^{|F|-1} \left(\frac{n_i - \tilde{n}}{\sigma} \right)^2} \quad (3.21)$$

where $|F|$ is the cardinality of the feature set, n_i is the number of attributes/dimensions between f_i and f_{i+1} , f_i is the i^{th} feature of the feature set, \tilde{n} is the number of attributes/dimensions expected between f_i and f_{i+1} if it is optimally spaced, and σ is the standard deviation (from Eqn. 3.23). Optimal distribution is defined as features that are equally spaced throughout the entire data domain. For example, a feature set that is $|F| = 6$ and has a data domain that has 100 attributes/dimensions would have an $\tilde{n} = 20$, which produces an optimal distribution resulting in $\iota = 0$, the best case distribution. The equation for \tilde{n} is:

$$\tilde{n} = \frac{N}{|F| - 1} \quad (3.22)$$

where N is the total number of attributes/dimensions across the data domain. The variance σ^2 is defined as:

$$\sigma^2 = \tilde{n} \left(1 - \frac{\tilde{n}}{N} \right). \quad (3.23)$$

As it is shown with the best case ι , optimally distributed where $\iota = 0$ and denoted as ι_{best} , the worst distributed case is determined by calculating the ι equation, so that each feature is consecutive. This equation for the worst case ι (ι_{worst}) is determined as shown:

$$\iota_{worst} = \sqrt{|F| - 1 \left(\frac{1 - \tilde{n}}{\sigma} \right)^2}. \quad (3.24)$$

The worst case, ι_{worst} , is used to provide a bounded range of distribution based on the dimension size of the data set and the size of the feature set. This bounding allows for a percentage of distribution to be defined by the user and is used in the final stage of the feature selection method of NASAFS-IDF.

In order to prevent the distributed spacing equation from producing inaccurate results for any combination of feature locations in the feature set, certain mathematical scalings are levied. If n_i is determined to be larger than \tilde{n} , then n_i will be assigned the value of \tilde{n} (i.e. $n_i = \tilde{n}$). This is necessary to prevent erroneous values of ι in cases where feature spacing does not warrant the ι value calculated. It is possible to have one or more n_i larger than \tilde{n} , which would produce a result larger than the worst case ι (ι_{worst}). Therefore, the above constraint is enforced to provide bounds and keep the ι value pertinent to the purpose of this problem. The following example provides a better understanding of this constraint. Consider a data domain of 100 dimensions, and a feature set size of six. For this case, $\tilde{n} = 20$, $\sigma = 4$, and $\iota_{worst} = 10.62$. If a specific feature set is picked with the features (5, 6, 7, 8, 9, 100), then the distributed spacing equation would return a value of $\iota = 20.13$. This is well beyond the value of the worst case (ι_{worst}), yet it is preferable to a worst case scenario, since at least one of the features is not adjacent to the rest of them. In order to give this feature set a relativistic value that can be compared to other feature sets, the above mathematical constraint ($n_i = \tilde{n}$) is imposed; therefore, for the distance between f_5 and f_6 , the value is $n_5 = 20$. This constraint produces a new value of $\iota = 9.50$. This new value is less than the worst case value (ι_{worst}) but close to the worst case, as would be expected based on the feature set.

3.2.1.3 Step 3: The Heuristic. The heuristic incorporates the two-fold approach, dependence and distance measures, in order to determine a feature set's discriminatory capability. This is similar to NASAFS; however, NASAFS-IDF differs from NASAFS in its incorporation of the distributed spacing function. The purpose of the distributed spacing function in NASAFS is to provide a cross-check

of the feature set, prior to its being evaluated by the heuristic. NASAFS-IDF uses its distributed spacing function within the heuristic, as an objective that is weighted and vectored along with the dependence and distance measures. This new heuristic value is scalarized and then optimized by the simulated annealing process. In cases where there are multiple objectives, a weighting scheme is typically used, in which all of the weights sum to 1 (i.e. $W_1 + W_2 + \dots + W_k = 1$). However, instead of performing a multi-objective optimization on the objectives of this heuristic, an assumption is made; it is assumed that the most important objective is the discrimination capability of the heuristic. It is necessary to be able to discriminate between classes, regardless of how the features are distributed across the domain. Selecting a feature set due to the optimization of the distribution of features, regardless of the discriminatory capability, would be counterproductive. Therefore, the heuristic of NASAFS-IDF, h_N , is:

$$h_N = A\gamma + h \tag{3.25}$$

where $A \in \{0, 0.5, 1\}$ is the weight applied to the result of the distributed spacing ratio γ , and h is defined in Equation 3.19. The distributed spacing ratio (γ) is a normalized difference of the worst case ι , Equation 3.26, and the actual ι of the current feature set. As the actual ι becomes more optimal (i.e. closer to 0), the value of γ approaches 1. Therefore, the distributed spacing objective is bounded between 0 and 1. The equation for γ is:

$$\gamma = \frac{\iota_{worst} - \iota_{actual}}{\iota_{worst}}. \tag{3.26}$$

The value of A can be one of three discrete values: 0, 0.5, or 1. These weight values are determined by the discriminatory value of the feature set. If the feature set has little or no ability to discriminate between classes, then $A = 0$. Subsequently, if it has moderate ability, then $A = 0.5$; if it has strong ability, then $A = 1$. In

essence, if the feature set is a poor discriminator, the overall value returned by the heuristic will not be improved, regardless of feature distribution. The value assigned for distribution will not decrease the value assigned for discrimination; however, it can be increased, depending on the distribution and its discriminatory capability. The equation for A is given by:

$$A = \begin{cases} 0 & \text{if } xcov_{x_{ref}, x_{tgt}} \geq k_h, \\ 1 & \text{if } xcov_{x_{ref}, x_{tgt}} < k_h \text{ and } D(R_{x_{ref}}, R_{x_{tgt}}) - d_h \geq 0, \\ 0.5 & \text{if } xcov_{x_{ref}, x_{tgt}} < k_h \text{ and } D(R_{x_{ref}}, R_{x_{tgt}}) - d_h < 0. \end{cases} \quad (3.27)$$

Where x_{ref} is the reference feature set, x_{tgt} is the target feature set, k_h is the covariance threshold, $D(R_{x_{ref}}, R_{x_{tgt}})$ is the absolute distance of the feature sets' autocorrelation values for the target and reference samples, and d_h is a distance threshold.

The heuristic of NASAFS-IDF performs a one-versus-all approach, instead of the pairwise process performed in NASAFS. This new process in NASAFS-IDF compares the covariance values of the specific feature set for all the classes being tested. The worst case value (the highest covariance value) of all these values is then selected as the covariance value; this is used to determine the appropriate scalar value of Eqns. 3.19 and 3.27. It is the feature set of the class that produced the highest covariance value that is used in determining the distance values in Eqn. 3.17. Comparing all classes simultaneously and choosing the worst case scenario allows for an accurate discrimination of classes and considerably reduces computational costs. The value calculated by the heuristic is sent to the simulated annealing process to determine feature set acceptance.

3.2.1.4 Steps 4-5: Feature Set Updates. After the first evaluation by the heuristic, the three stage process (*selection, evaluation, and candidacy determination*) repeats. One feature within the selected feature set is replaced by a feature that is randomly selected from the set of available features. The resulting feature

set is then evaluated by the heuristic, as is accomplished according to step 3 (Section 3.2.1.3), and the value is then returned to the simulated annealing process for candidacy determination. This process is repeated until convergence; at this point, the current feature set is the solution.

3.2.2 Final Feature Selection Process. Since NASAFS-IDF is a stochastic process, Monte Carlo simulations are performed and a histogram exists for the reference class versus all other classes. Therefore, this part of the NASAFS-IDF method is to determine *the* feature set from the histogram that provides the best classification possible, based on the percentage of optimal distribution desired by the user. The variables H_{Hit} , and A_{val} are no longer needed.

The values of the histogram are ranked from highest to lowest; the feature set is picked from this order, based on the percentage of distribution specified. The highest ranked feature is chosen, and becomes a feature of the feature set. Then the next highest feature is chosen; however, it must be a feature that exists outside of a specified area that is centered on each feature of the feature set. This area is determined by a distance measure (S) and is stipulated by the percentage of acceptable distribution. This distance is given by:

$$S = \left\langle \tilde{n} - \left[\sigma \left(\sqrt{\frac{(\iota_{allowed})^2}{|F| - 1}} \right) \right] \right\rangle \quad (3.28)$$

where \tilde{n} is the expected number of attributes/dimensions between f_i and f_{i+1} if optimally spaced (Eqn. 3.22), σ is the standard deviation given by Eqn. 3.23, $\langle \cdot \rangle$ is the round function, $|F|$ is the cardinality of the feature set, and $\iota_{allowed}$ is the allowed distributed spacing based on the percent of acceptable distribution and is calculated as:

$$\iota_{allowed} = \left(1 - \left[\frac{\%_{acceptable}}{100} \right] \right) \iota_{worst} \quad (3.29)$$

where $\%_{acceptable}$ is the user-defined percentage of acceptable spacing desired, and ι_{worst} is defined by Eqn. 3.24.

Once an attribute is selected from the histogram to be a feature for the feature set, the remaining attributes within the defined distance S around the selected feature (in the histogram) are occluded from selection. This process is shown in Fig. 3.10. In Fig. 3.10, the highest ranking feature is determined (1^{st} feature picked), and is indicated in red; the attributes that are no longer viable options are occluded, as shown with the green box. After these attributes are eliminated, the next largest attribute is selected from the remaining attributes; this attribute then becomes the next feature for the feature set, and it is denoted in Fig. 3.10 as 2^{nd} feature picked. The surrounding attributes that are within the defined distance are then occluded, as indicated by the green box around the 2^{nd} feature picked. This process of feature selection and subsequent attribute occlusion repeats until the feature set is filled. Figure 3.11 shows the histogram created for the texture data set, processed with a 35% acceptable distribution and a feature set size of six; the dotted line indicates the attributes selected and their order of selection. The NASAFS-IDF pseudo-code is shown in Algorithm 3.

Fig. 3.12 illustrates the acceptable spacing technique. For example, if the domain is 100 dimensions and the feature set size is six, then optimal spacing occurs when there are exactly 20 dimensions between each feature, as shown in the top number line of Fig. 3.12. If instead an 80% acceptable spacing is chosen, there are multiple possible solutions; one possible solution is shown in the number line at the bottom of Fig. 3.12, where the distance spacing (S , from Eqn. 3.28) for this particular example is 16. Therefore, the acceptable distribution is variable, dependent on the percent acceptable distribution determined by the user. The percent acceptable distribution may also be limited by the data set and the size of the feature set desired.

Algorithm 3: Non-correlated Aided Simulated Annealing Feature Selection
- Integrated Distribution Function (NASAFS-IDF) pseudo-code

Input: Data set of samples with labels and user defined variables(udv)
Output: Feature set for a class versus all other classes

INITIALIZATION
begin
 // Bin data, see Fig. 3.1 and 3.3
 DataBinned \leftarrow bin data based on BinSize(udv)
 // Compute cross-covariance threshold
 for $j = 1 \dots \text{number of bins}$ **do**
 for $i = 1 \dots \text{number of reference samples}$ **do**
 $k_h[j, i] \leftarrow \text{xcov}(x_{j,i}, x_{j,K})$
 $k_{h(\min)} \leftarrow$ minimum of k_h
 // Monte Carlo Simulation
 while Monte < NumRuns(udv) **do**
 // Select initial feature set F
 F \leftarrow Randomly select NumFeatures(udv) new features
 Convergence \leftarrow No // Initialized to 'no'
 while Convergence = no **do**
 EVALUATION
 begin
 // heuristic evaluates feature set, determine h
 $D_h \leftarrow 0.002$ // Initialized distance threshold
 if ($\text{xcov}(x_{ref}, x_{tgt}) \geq k_{h(\min)}$) or ($\text{xcov}(x_{ref}, x_{tgt}) < k_{h(\min)}$ and
 $\text{Dist}(R_{x_{ref}}, R_{x_{tgt}}) < D_h$) **then**
 $h \leftarrow 1 - \text{xcov}(x_{ref}, x_{tgt})$
 if ($\text{xcov}(x_{ref}, x_{tgt}) < k_{h(\min)}$) and ($\text{Dist}(R_{x_{ref}}, R_{x_{tgt}}) \geq D_h$) **then**
 $h \leftarrow |\text{xcov}(x_{ref}, x_{tgt})| + 1 - \text{xcov}(x_{ref}, x_{tgt})$
 $D_h \leftarrow$ increased
 // R is the aurocorrelation and Dist is the Manhattan distance
 $A \leftarrow \in \{0, 0.5, 1\}$ // depends on h, Eqn. 3.27
 $\gamma \leftarrow$ distribution value, see Eqn. 3.26 and 3.21
 $h_N \leftarrow A\gamma + h$
 CANDIDACY DETERMINATION
 begin
 SA $\leftarrow h_N, T$ // see SA pseudo code Alg. 1
 T $\leftarrow T \times \text{schedule}$ // decrements T via schedule
 SELECTION
 begin
 // Select a new feature set
 F \leftarrow Replace one feature with a randomly picked feature
 MasterHisto \leftarrow current F // Histogram of feature sets
 NumRuns \leftarrow decrement NumRuns
 $F_{master} \leftarrow$ top feature of MasterHisto
 S \leftarrow required distance based on acceptable distribution, Eqn. 3.28
 for $u = 1 \dots \text{NumFeatures} - 1$ **do**
 $F_{master} \leftarrow$ next top feature not occluded by distance 'S' around previously selected feature,
 Section 3.2.2

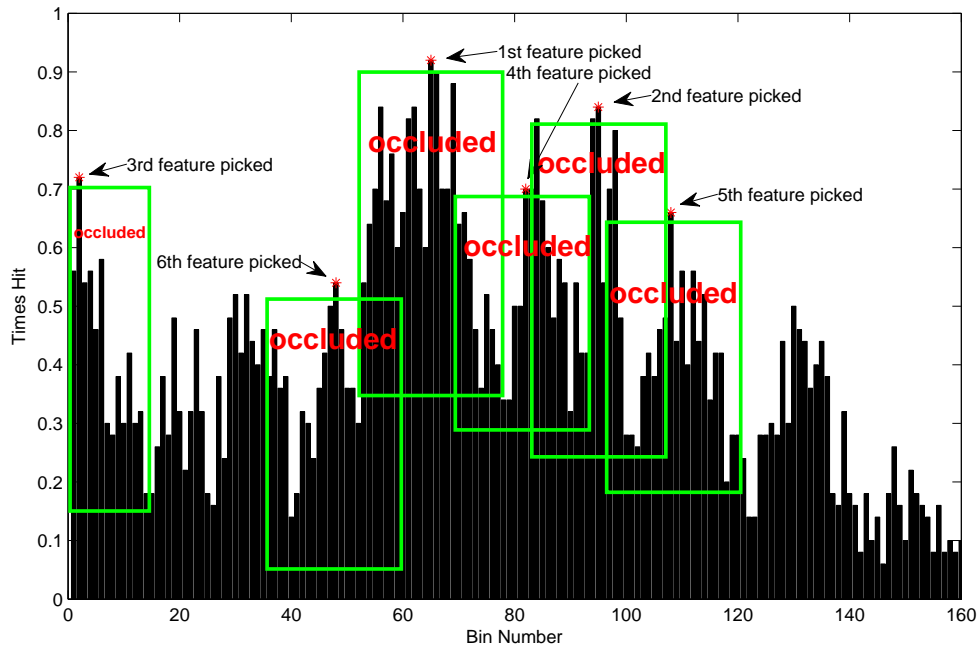


Figure 3.10: This is an example of the final feature selection process for 100% Cotton Woven, with acceptable distributed spacing set to a 35% optimal distribution. The order of feature assignment is indicated; green boxes indicate the attributes that are occluded due to feature selection.

3.3 Correlation Detection Method

The correlation detection method (CoDeM) is a classification method which uses the same principle of statistical evaluation of feature sets as both NASAFS and NASAFS-IDF. CoDeM calculates the accuracy results, and allows for the addition of additive white Gaussian noise, if desired. CoDeM performs detection by labeling a sample as either in-class or out-of-class, as compared to the reference sample; an out-of-class designation simply indicates the sample is not the same as the reference class. CoDeM uses the average value of each feature for its calculations, where a feature is a bin that can contain one or more attributes. Fig. 3.13 illustrates the feature averaging technique; in this diagram, the bin size is three and the feature set size is four. A flow chart showing the process of CoDeM is shown in Fig. 3.14. Once the data is obtained from the feature selection method, CoDeM loads the features (designated by the feature set) of the noisy reference samples and the features of

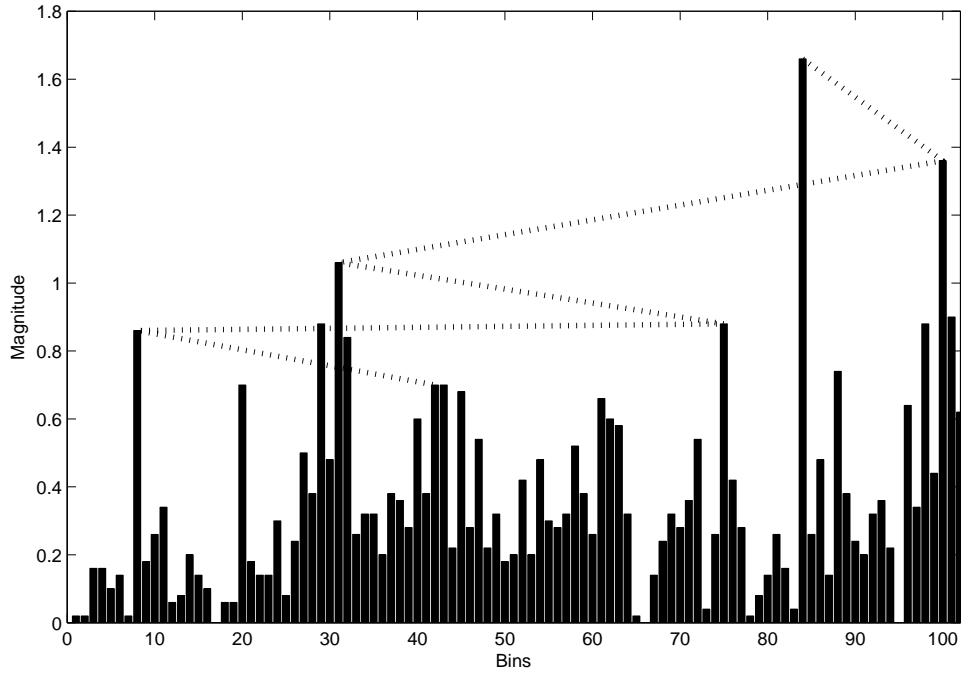


Figure 3.11: Histogram created by NASAFS-IDF, and used by final feature selection stage, to determine a feature set for the reference class. The dashed line indicates feature selection. This histogram is of the first class of the texture data set.

the single noisy average sample; CoDeM then sends them to the training function to obtain a covariance threshold k_{CoDeM} and a distance threshold d_{CoDeM} . It determines the worst cross-covariance (lowest) using every combination of the reference samples (as many samples as the user allows it to train with); this is accomplished for each corresponding feature of the feature set of the reference class.

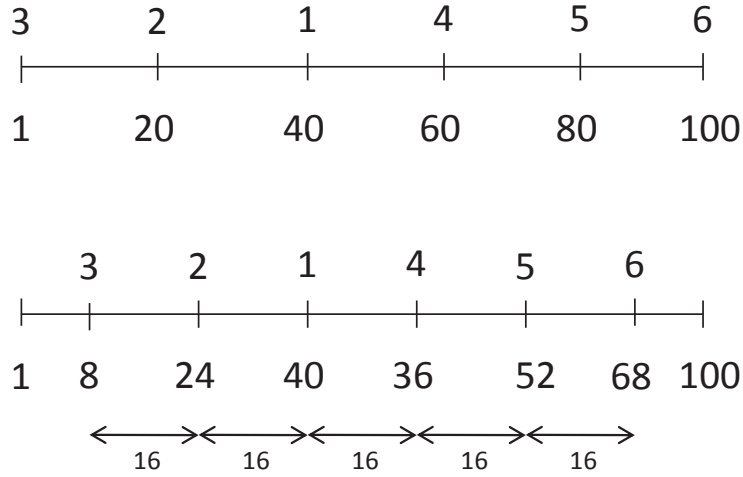


Figure 3.12: The top and bottom number lines indicate the domain of a sample with 100 dimensions. The numbers above each number line indicate the location of the i^{th} feature. The top number line indicates placement of features for acceptable distribution, whereas the bottom number line indicates one possible placement for an 80% acceptable distribution. The distance of 16 indicates the minimum number of dimensions between features for this percent acceptable distribution. The feature set size for both cases is 6.

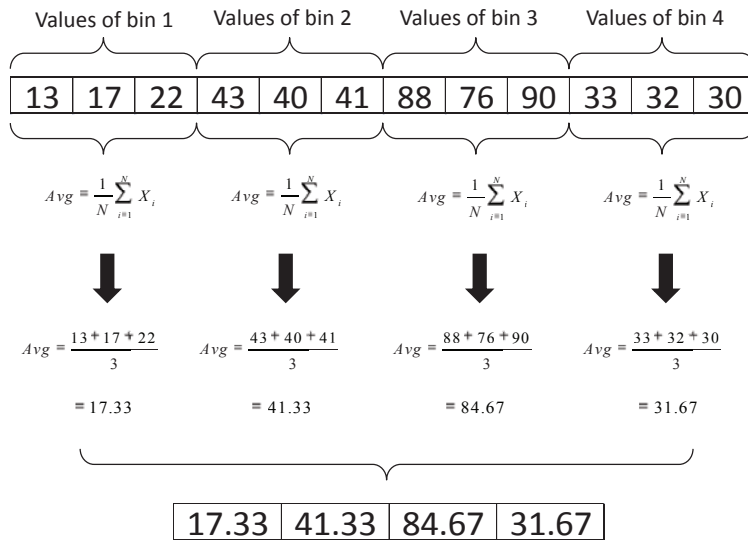


Figure 3.13: This is an example of the averaging technique for the CoDeM process.

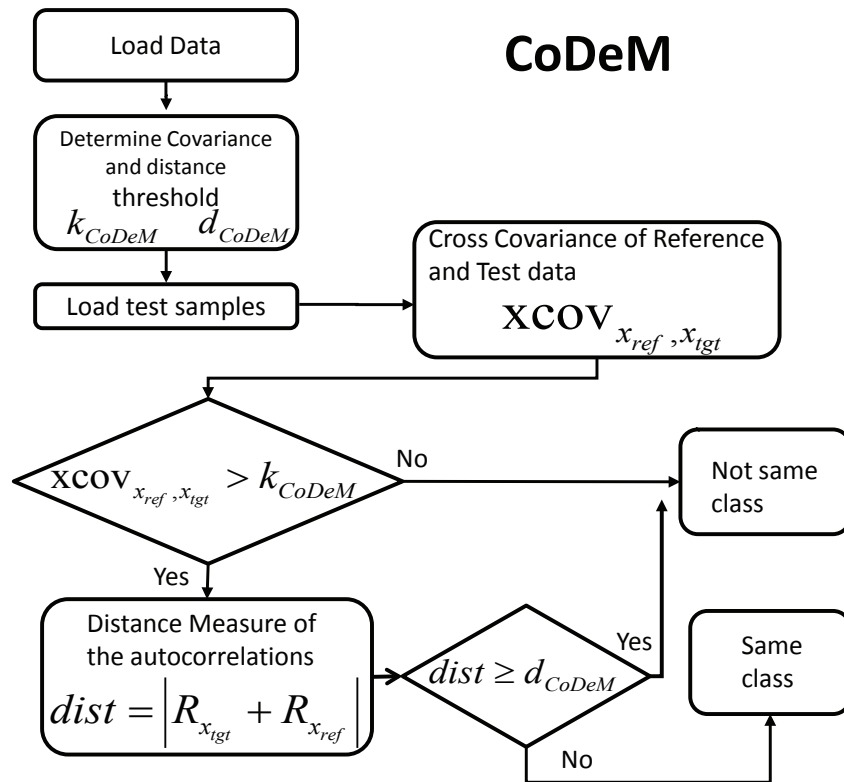


Figure 3.14: Flow chart for the detection algorithm Correlation Detection Method (CoDeM).

CoDeM loads every test class sample and adds noise to each one, as designated by the noise power value; it then separates out only the features as designated by the reference class feature set and takes the mean of each feature (Fig. 3.13). The cross-covariance $xcov_{x_{ref},x_{tgt}}$ is calculated from the mean of the clean reference class features to the mean of the noisy target class features.

To account for the noise and calibration differences during data collections that can lead to inaccurate classifications, the distance threshold (d_{CoDeM}) is calculated using a random sampling of the reference samples. The autocorrelations of both the clean reference sample feature set and the average noisy reference sample feature set are used to determine the distance threshold. If the current distance is greater than the previous distance, the threshold is updated by adding a fraction of the new distance to the previous distance, as shown:

$$d_{CoDeM} = \begin{cases} d_{CoDeM} \\ \text{if } d_{CoDeM} > d_{i+1}, \\ d_{CoDeM} + 1/2 (d_{i+1}) \\ \text{if } d_{CoDeM} \leq d_{i+1}, \end{cases} \quad (3.30)$$

where d_{CoDeM} is initially set to the first distance threshold value, and d_i is the distance represented as: $d_i = |R_{x_{ref_i}}(\text{clean}) - R_{x_{ref}}(\text{noisy avg})|$, (noisy avg) refers to the average of the noisy reference samples, $i = 1, 2, \dots$, number of samples chosen, $R_{x_{ref}}(\text{clean})$ is the autocorrelation of the clean reference class feature set and $R_{x_{ref}}(\text{noisy avg})$ is the autocorrelation of the average noisy reference class feature set.

The cross-covariance ($xcov_{x_{ref},x_{tgt}}$) is calculated from the mean of the features of the clean reference class feature set to the mean of the features of the noisy target class feature set. The absolute distance ($dist$) from the autocorrelation of the clean

reference class feature set to the noisy target class feature set is calculated as ($dist = |D(R_{x_{ref}}, R_{x_{tgt}})|$). Classification is determined as:

$$Class(X_i^C, y) = \begin{cases} \text{out-of-class} \\ \text{if } xcov_{x_{ref}, x_{tgt}} \leq k_{CoDeM}, \\ \\ \text{out-of-class} \\ \text{if } xcov_{x_{ref}, x_{tgt}} > k_{CoDeM} \text{ and} \\ dist \geq d_{CoDeM}, \\ \\ \text{in-class} \\ \text{if } xcov_{x_{ref}, x_{tgt}} > k_{CoDeM} \text{ and} \\ dist < d_{CoDeM}. \end{cases} \quad (3.31)$$

where X_i^C is the i^{th} sample of the S^{th} class, and y is either 1 or 0 for in-class or out-of-class, respectively. This process is expected to produce the most realistic results, if an appropriate target sensor noise model is incorporated [17, 77].

3.4 Summary

This chapter detailed the processes of the novel feature selection methods, NASAFS and NASAFS-IDF. Both methods determine an accurate feature set by using a stochastic method and by selecting features that are non-redundant. The two methodologies take different approaches, yet both produce excellent results. The detector (CoDeM) is described in detail; it is based on the same methodology used in both of our novel feature selection methods.

IV. Experimental Results and Analysis

NASAFS and NASAFS-IDF are compared to ReliefF [48], GRLVQI [61], and the Bhattacharyya feature selection methods [5]. The accuracy of the feature selection methods are evaluated using the correlation-based (CoDeM) detector, the Minimum Euclidean Distance (MED), Naïve Bayes [33], and C4.5 classifiers [73], as in [38]. Three different data sets are used in the evaluation of the above feature selection methods. The first is a 12 class hyperspectral textile data set, shown over a range of additive white Gaussian noise realizations. The second is the Lunar Crater Volcanic Field data set (LCVF), a VI/NIR/SWIR(0.4–2.5 μm) 224-band, 30 m/pixel Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) image of the Lunar Crater Volcanic Field in Nevada [10, 61, 62]. The third is a 7 class image texture data set taken from the Brodatz samples and preprocessed using a 5 level wavelet decomposition [87].

The results are compared using contingency tables, correlation coefficients of the feature sets, and overall accuracies, combined with the significance of the feature selection methods, as defined by the Wilcoxon signed-rank test. Other comparisons, such as the Pareto front, are provided to present a complete picture of the abilities of NASAFS and NASAFS-IDF.

4.1 Comparison Tests

The contingency table is a tool used to compile the success of a classifier. From the contingency table, five important statistical evaluations can be obtained: the *kappa statistic*, *commission error*, *omission error*, *producers accuracy*, and *consumers accuracy* [23]. The kappa statistic provides a means to evaluate the classifier's ability to predict the classification of the samples as compared to a random estimation of their classification. If the kappa statistic is close to zero, then it suggests that the results are most probably achieved by chance; if the result is closer to one, then it suggests that the results are from accurate classification, not estimation. Commission error is defined as the percentage of samples that are incorrectly classified, meaning that they are identified as belonging to a class other than their actual class. Omission error is

the percentage of samples that belong to a certain class, but were not classified as belonging to that class. Producers accuracy is the percentage which shows the number of correctly classified members of a class. Consumers accuracy is the percentage which shows the number of samples correctly identified as belonging to a class, out of the total number of samples identified by the classifier as belonging to that class. The overall accuracy of the contingency table is the sum of the accurately classified samples per class (calculated by taking the sum of the diagonal) divided by the total number of samples used. As an example, a contingency table is shown in Fig. 4.14; it was generated for the 12 class textile data set using Naïve Bayes, with 30% acceptable spacing.

The Wilcoxon signed-rank test is a non-parametric statistical method used to compare two similar measures or methods, typically when the data sets are not normally distributed [82]. It produces a test statistic, which can then be compared to a critical values table. The result of this comparison is used to determine the level of significance desired, based on rejection of the null hypothesis. By definition, the null hypothesis cannot be validated; it can only be rejected or not rejected. The alternative hypothesis is contradictory to the null hypothesis; it is validated upon rejection of the null hypothesis. For example, for our purposes, the null hypothesis is defined as *method A is not significantly better than method B*; the alternative hypothesis is *method A is significantly better than method B*. In this example the hypotheses establish that the critical table to be used should be a one-tailed table; this is determined by the directionality of the hypotheses. Based on this table, the level of significance is then deduced and the null hypothesis is either rejected or not rejected [82,92].

In multi-objective optimization problems, there are typically several solutions to the problem, based on the objectives to be optimized. Of these solutions, some are better than others, when regarding all the objectives considered; however, a set of solutions exist in which no solution is completely better or worse than another solution. If a solution cannot be *completely* dominated by another solution (i.e. no solution is a better solution for all objectives concerned), it is part of the non-dominated solution

set. In this solution set, all solutions are equally viable; some solutions may be better than others at achieving a particular objective, but all the solutions are acceptable. The non-dominated solution set can be compiled into a list; this list is termed the Pareto Front [83, 84]. Any of the Pareto Front solutions can be acceptable solutions; acceptability being based on the user-determined desired results. In this work, accuracy and feature set size were the objectives determined for optimization. Fig. 4.13 shows an example of the Pareto Front generated for the 12 class textile data set; this is achieved using CoDeM and a 30% acceptable distributed spacing.

4.1.1 Data. The high spectral resolution 12 class textile data set contains 1600 dimensions and was collected by a hand-held reflectometer with a sampling interval of $1nm$. Due to the parameters, spectral mixing is not considered in this classification problem. NASAFS and NASAFS-IDF are processed using a bin size of $10nm$, which is the average bandwidth of a typical hyperspectral imaging collection system (e.g. AVIRIS). Therefore, in order to perform a fair comparison, the high-resolution data ($1nm$ sampling interval data) is resampled (averaged) to $10nm$ for ReliefF, GRLVQI, and the Bhattacharyya methods. The classes of the 12 class textile data set are shown in Table 4.1, with representative samples shown in Fig. 4.1.

Table 4.1: Class types of the 12 class textile data set.

Class	Type	Constituents	Fabric Class	Samples
1	<i>Blend</i>	65% <i>Polyester</i> 35% <i>Cotton</i>	<i>Woven</i>	18
2	<i>Blend</i>	80% <i>Nylon</i> 20% <i>Spandex</i>	<i>Knit</i>	18
3	<i>Blend</i>	80% <i>Polyester</i> 20% <i>Rayon</i>	<i>Woven</i>	18
4	<i>Blend</i>	94% <i>Polyester</i> 6% <i>Spandex</i>	<i>Woven</i>	18
5	<i>Blend</i>	97% <i>Bamboo</i> 3% <i>Spandex</i>	<i>Knit</i>	18
6	<i>Blend</i>	97% <i>Cotton</i> 3% <i>Spandex</i>	<i>Woven</i>	18
7	<i>Pure</i>	100% <i>Cotton</i>	<i>Knit</i>	18
8	<i>Pure</i>	100% <i>Cotton</i>	<i>Woven</i>	18
9	<i>Pure</i>	100% <i>Polyester</i>	<i>Knit</i>	18
10	<i>Pure</i>	100% <i>Polyester</i>	<i>Woven</i>	18
11	<i>Pure</i>	100% <i>Satin</i>	<i>Woven</i>	18
12	<i>Pure</i>	100% <i>Silk</i>	<i>Woven</i>	18

The low spectral resolution Lunar Crater Volcanic Field data set (LCVF) is a VI/NIR/SWIR(0.4–2.5 μm) 224-band, 30 m/pixel Airborne Visible/Infrared Imaging

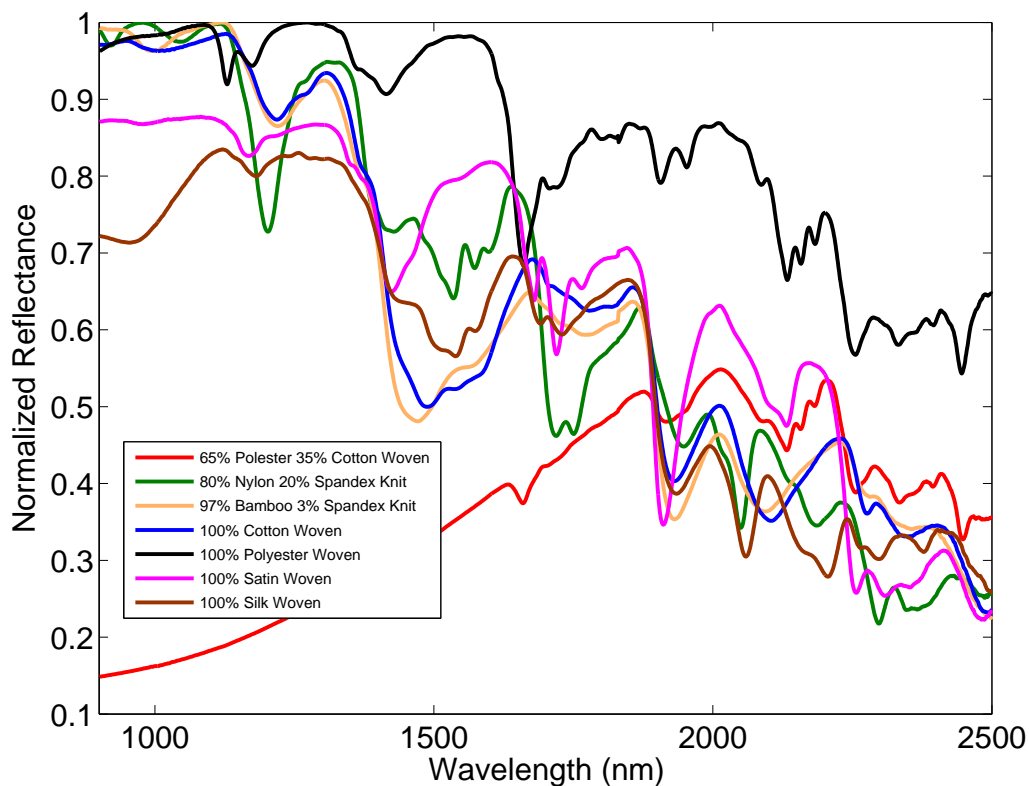


Figure 4.1: Representative samples from the 12 class textile data set: 65% Polyester 35% Cotton Woven (red), 80% Nylon 20% Spandex Knit (green), 97% Bamboo 3% Spandex Knit (tan), 100% Cotton Woven (blue), 100% Polyester Woven (black), 100% Satin Woven (pink), and 100% Silk Woven (brown).

Spectrometer (AVIRIS) image of the Lunar Crater Volcanic Field in Nevada, USA collected in 1994 [10,61,62]. The LCVF data set is correlated between the classes and presents a challenging problem. Fig. 4.2 shows representative spectra samples from the 23 class LCVF data set used in this paper [61]. On average, the LCVF data set was collected with a $10nm$ sampling interval; this qualifies it, for the purposes of this work, as low-resolution data.

The 7 class Brodatz image data set is a 640×640 pixel GIF data set [87]. Each image is divided into 12 sample images that are 241×241 images with overlapping edges. The 7 textures used are shown in Fig. 4.3, along with their corresponding class number. These samples are preprocessed using a two-dimensional 5^{th} level wavelet

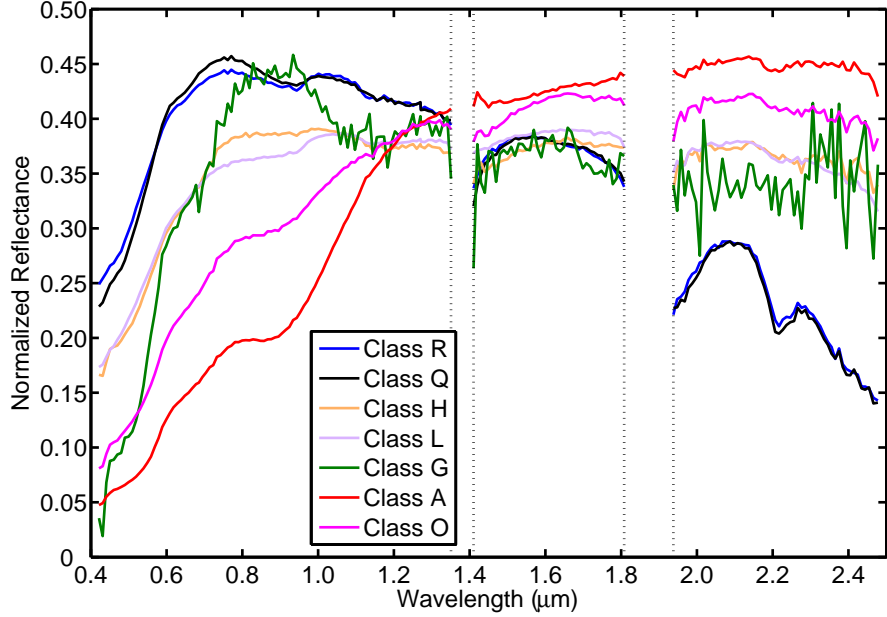


Figure 4.2: Select representative spectra of the Lunar Crater Volcanic Field (LCVF) 23 class data set, classes A (red), G (green), H (orange), L (magenta), O (purple), Q (black), and R (blue). The water absorption bands are indicated by the vertical dotted lines [61, 62].

decomposition [86]. The bins of the 5th level (called leaves) are processed via the entropy calculation; this results in the vectorization of the 5th level leaves. This vector processing produces a sample that is 1024 dimensions long; the processing is then repeated for each image. Appendix A provides further discussion of the wavelet decomposition method.

4.1.2 Configurations of Test. The feature selection methods tested in this work are implemented using a three-fold cross-validation. A larger k-fold cross-validation could not be used, due to the sample sizes of most of the data sets. For all of the feature selection methods employed in this work, no more than six features are allowed to be in the final feature set for comparison between methods. The selection of six features is arbitrary and used for computational considerations; however, trade studies of different feature set sizes were conducted and the results are reported.

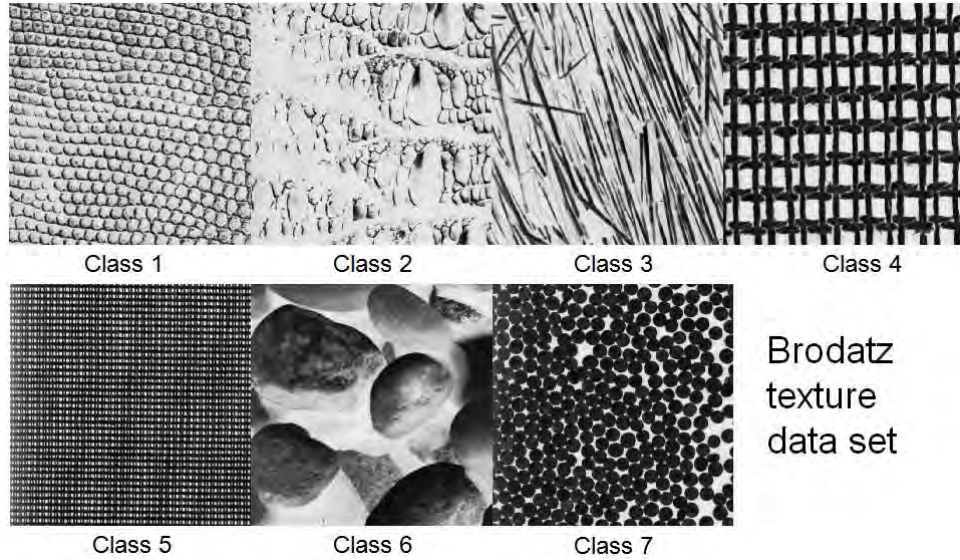


Figure 4.3: Brodatz samples with their associated class labels [87].

For the 12 class textile data set, four different noise realizations are added within the CoDeM detector, in order to gain insight into the robustness of the feature sets chosen. Noise power values are chosen based on the average noise levels of a typical fielded imaging system; for the purposes of this work, we used the SpectTIR’s HyperSpectTIR V to compute noise values [40]. The noise is standard unit variance additive white Gaussian noise that is added to the signals, based on a noise power level. The noise power levels used are 0.0000, 0.0125, 0.0250 and 0.0300. Fig. 4.4 illustrates the difference between a clean hyperspectral signal and a hyperspectral signal with additive white Gaussian noise of a power level of 0.03. CoDeM also adds a fraction of the added test data noise to the reference data, in an attempt to create a more accurate detector. This is only performed for lab quality (e.g. closed system) data; if the data is collected in the field, no noise is added, as it is already representative of a noisy system and environment.

The average correlation coefficient is computed for each feature set of each feature selection method evaluated. The average correlation coefficient for each feature set is obtained by averaging the correlation coefficients for all of the combinations of each feature within the feature set (Fig. 4.5).

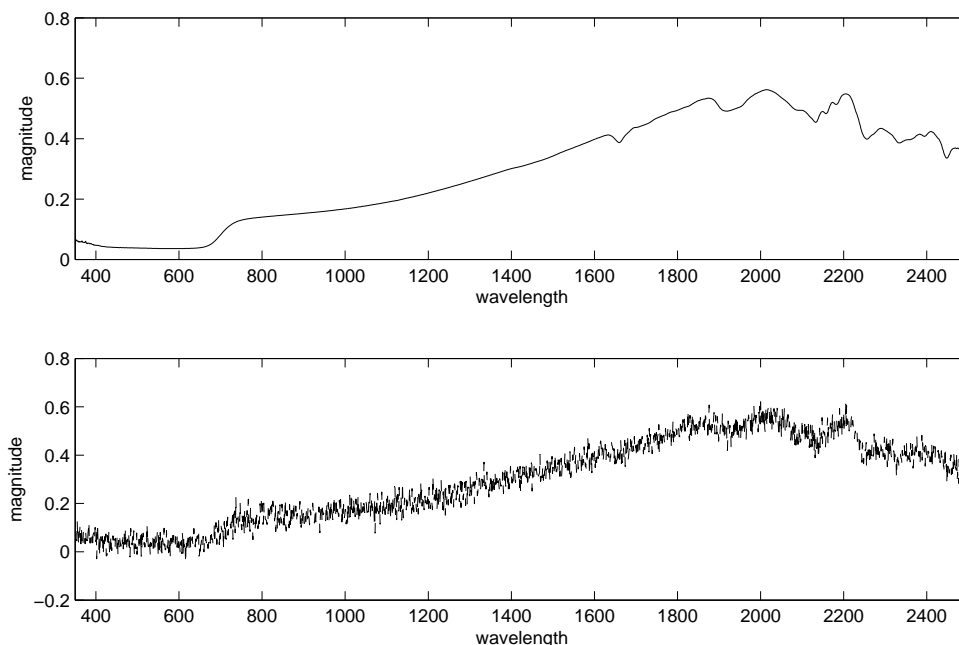


Figure 4.4: Diagram of two hyperspectral signals. The top figure depicts a clean hyperspectral signal of 65% Polyester 35% Cotton blend; the bottom figure depicts the hyperspectral signal for 65% Polyester 35% Cotton blend, with additive white Gaussian noise of a 0.03 power level.

ReliefF, GRLVQI, and the Bhattacharyya produce global feature sets, whereas NASAFS and NASAFS-IDF do not. Therefore, the accuracies obtained from the classifiers for both novel feature selection methods, NASAFS and NASAFS-IDF, with CoDeM and MED are averages of the class accuracies, where Naïve Bayes and C4.5 are based off of the adjusted contingency table explained in Section IV. Due to CoDeM’s processing specifications, its results are class averages for all feature selection methods. ReliefF, GRLVQI, and the Bhattacharyya methods produce global feature sets; therefore, the global feature sets are used in the same manner as if they were class feature sets when classified by CoDeM. The accuracies are then calculated the same as they are for NASAFS.

For NASAFS, the correlation matrix is computed using all available samples; this allows for the distributed spacing function to use the determined sub-regions in

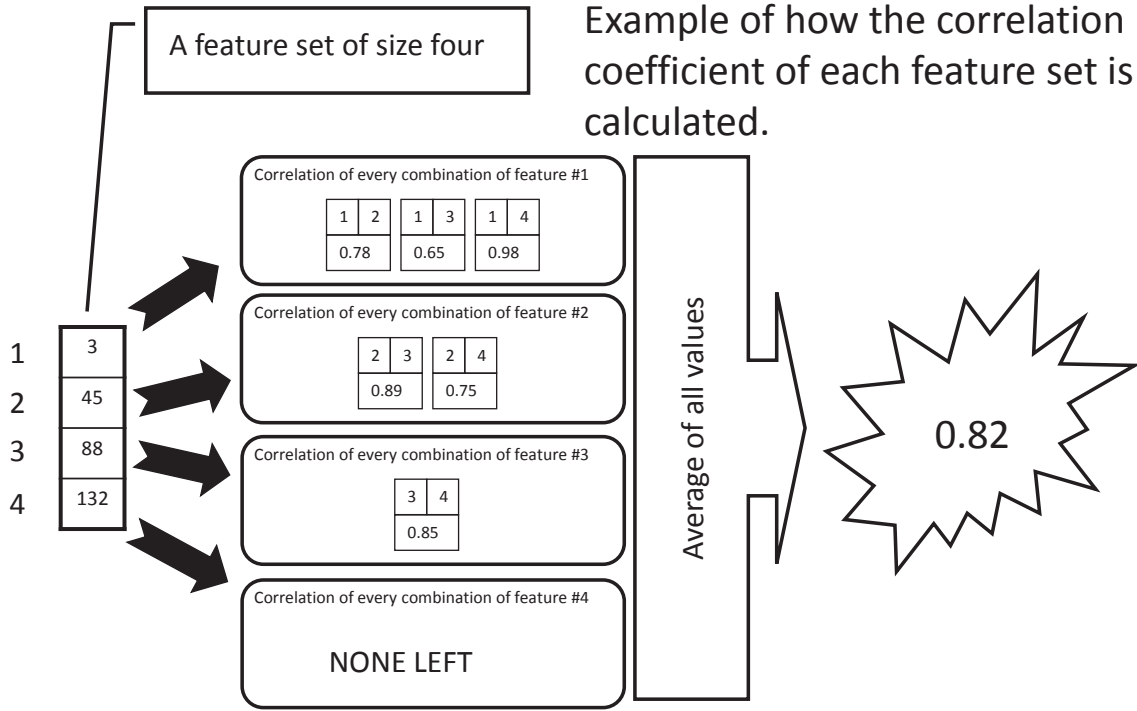


Figure 4.5: Pictorial diagram showing the process used to obtain the correlation coefficient value for each feature set.

the spectral domain. The spectral domain is divided into five sub-regions based on the correlation matrix for the 12 class textile data set. It is divided into four sub-regions for the LCVF data set. The texture data set was not tested in NASAFS; therefore, no sub-regions are calculated.

4.2 NASAFS & NASAFS-IDF Parameters

There are several parameters that affect the performance of NASAFS and NASAFS-IDF. These parameters include: the initial temperature T_{int} , and the final temperature T_{final} of the simulated annealing algorithm, the decay rate, T_{decay} , of the temperature variable (T), the bin size N_{bin} , minimum acceptable number of times a feature is selected across all histograms (H_{hit}), minimum acceptable number of times that a feature is selected within a histogram (A_{val}), the noise power used for CoDeM NdB , the acceptable distributed percentage $APct$, and the feature set size ($|F|$).

NASAFS and NASAFS-IDF are implemented using a bin size of $10nm$, which is the average bandwidth of an imaging collection system (e.g. AVIRIS). Bin size helps both methods locate specific features within the spectral domain of the signal; these features exploit the underlying structure in the high-resolution data, and target the capability of the low-resolution collection system. Different bin sizes can be specified to determine the best sampling for a target system. For one instance, a bin size of 4 ($4nm$) may be able to find better discriminating feature sets than a bin size of 40 ($40nm$); in another instance, a bin size of 4 may be too restrictive to locate the best feature sets based on the data, and a bin size of 40 may be more appropriate. Ultimately, the bin size is set to the collection system's specifications; this allows NASAFS and NASAFS-IDF to find the best discriminating feature set, based on the target collection system's capabilities.

Some parameters have an obvious effect on the effectiveness of the algorithm. Two of these parameters are temperature decay rate and final temperature. Adjustment of the temperature decay rate can affect the number of poorly discriminating feature sets that are accepted. Decay rate and feature set acceptance are inversely proportionate. As the decay rate increases, the number of bad (undesirable) feature sets accepted decreases; as the decay rate decreases, the number of bad feature sets accepted increases. Finding an acceptable equilibrium for the temperature decay rate is necessary to optimize the efficiency and completeness of the algorithm. Final temperature is a parameter that is a factor in the temperature decay rate function. Setting the final temperature too low decreases efficiency; setting the final temperature too high could cause the algorithm to terminate prior to convergence. This value was set based on an empirical study of different starting temperatures with decay rates until an acceptable amount of chance selects were allowed.

Convergence is the point at which the simulated annealing process completes a feature set and therefore terminates. At this point, no new feature can be added to the feature set to achieve a better feature set. At present, determining a good convergence value is a matter of observing the Monte Carlo runs; the empirical results obtained

can then be used to gain insight into that value. The convergence value is currently established at 75 iterations, with no improvement to the previous feature set.

Starting temperature is another parameter that affects efficiency of the algorithm. The higher the starting temperature, the greater the chance of accepting a bad feature. Therefore, a good starting temperature must be determined, in order to prevent too many bad features from being randomly added. Through experimentation, it has been established that a starting temperature of ≈ 0.4 is acceptable. This is due to the fact that the exponential equation is dependent on the average *err* value, which is dependent on the heuristic value returned, as seen from the exponential equation ($\exp(\frac{err}{T})$). Fig. 4.6 illustrates the number of chance selects, based on the initial temperature of the simulated annealing process; starting temperatures of 0.02, 0.4, and 1.0 are used. The temperature is decayed by 10% for each iteration. The diagrams in Fig. 4.6 consist of 100 Monte Carlo runs, where each run produced about 210 iterations, based on its final temperature and the convergence criteria. Each Monte Carlo run has a set of randomly generated values between 0 and 1; this simulates the values returned by the heuristic. Each diagram in Fig. 4.6 uses the same randomly generated data for each run and its exponential is compared with the same random number for determination of value retention. For the starting temperature of $T = 0.4$ (left figure of Fig. 4.6), on the average, chance selection ceased after 20 iterations, averaging 3 chance selects per Monte Carlo run, for a total of approximately 300 chance selects. Using a starting temperature of $T = 1.0$ (middle figure of Fig. 4.6), on the average, chance selection ceased after 30 iterations, averaging 7 chance selects per Monte Carlo run, for a total of approximately 700 chance selects. For a starting temperature of $T = 0.02$ (right figure of Fig. 4.6), on the average, chance selection ceased after 3 iterations, averaging 0.2 chance selects per Monte Carlo run, for a total of approximately 24 chance selects. While a starting temperature of $T = 1.0$ and $T = 0.4$ averaged approximately the same number of minimum iterations before chance selections ceased, $T = 1.0$ allows a significantly higher number of chance selects to occur than does $T = 0.4$. Therefore, it is evident that, if the starting temperature is too

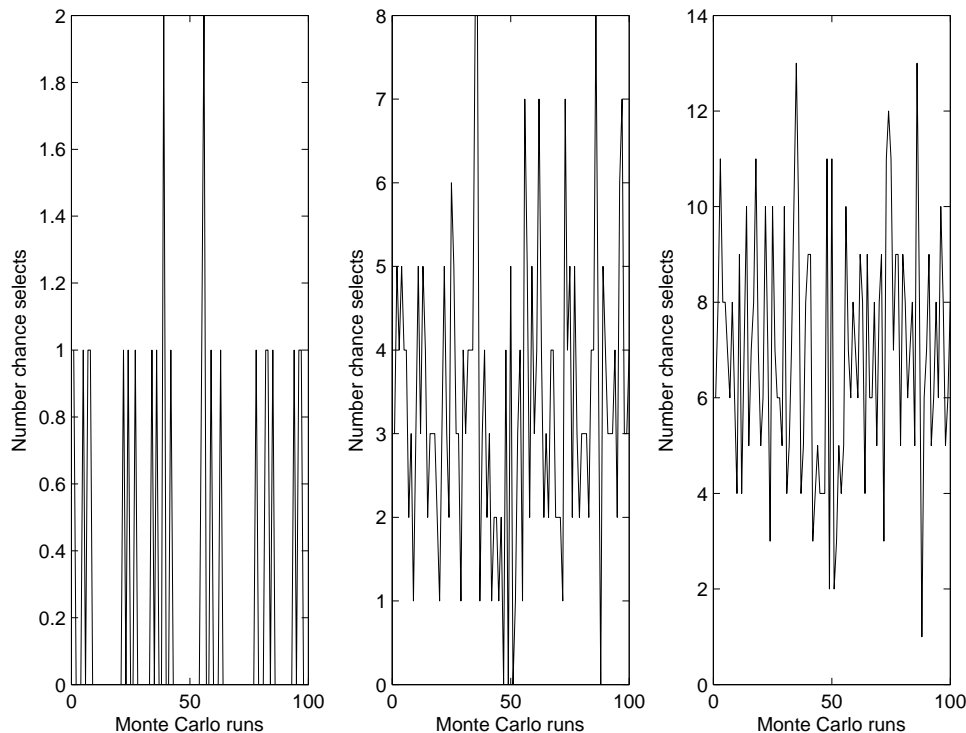


Figure 4.6: Illustration of the chance selects, based on the initial temperature of the simulated annealing process. Diagrams show results (left) using an initial temperature of 0.02, (mid) using an initial temperature of 0.4, and (right) using an initial temperature of 1.0.

high ($T = 1.0$), the process accepts too many bad chance feature selects, increasing the possibility that it could miss a significant global maximum/minimum; conversely, if the starting temperature is too small ($T = 0.02$), the process does not generate enough chance selects, severely limiting the total number of chance selects and increasing the probability that the process will settle on a local maximum/minimum. Therefore, a moderate starting temperature of $T = 0.4$ is used in this work. The experiments of this work use the parameters in Table 4.2, unless otherwise stated.

4.3 Results of Experiment

4.3.1 *Textile Data Results.* NASAFS and NASAFS-IDF successfully select features that are distributed across the spectral domain and that have good discrim-

Table 4.2: NASAFS and NASAFS-IDF feature selection parameters.

Parameter	Value	Description
T_{int}	0.4	starting temperature
T_{decay}	10%	temperature decay rate
T_{final}	1×10^{-10}	final temperature

inating characteristics. Fig. 4.7 shows the feature sets selected for four runs of the NASAFS method, comparing 65% Polyester 35% Cotton Blend (class 1) to 80% Nylon 20% Spandex Blend (class 2). The feature sets' average correlation coefficient is low, as show in Table 4.3. The covariance can be used to obtain a measure of discrimination between the feature sets. Calculating the covariance of the feature set between class 1 and class 2 for each of these runs results in a low covariance value that ranges from -0.9401 to -0.9991, which indicates good discrimination between the classes; it can be determined from this, that one vector is increasing as the other is decreasing. NASAFS and NASAFS-IDF feature selection methods are stochastic processes, each incorporating a random walk; therefore, a Monte Carlo simulation is used, in order to create a histogram of the feature sets. This allows the final feature selection stage to identify the best feature set from the histograms created, as shown in Fig. 3.8 and 3.11. This stage selects the histogram feature set that best discriminates these two classes, based on the distributed spacing constraint (Section 3.1.2 and 3.2.2). The correlation matrix used for NASAFS is shown in Fig. 3.4, where the horizontal black lines indicate the different sub-regions chosen by NASAFS.

Table 4.3: Average correlation coefficients of four single runs of NASAFS comparing 65% Polyester 35% Cotton blend to 80% Nylon 20% Spandex blend.

Run	Mean Corr Coef
1	0.5100
2	0.4809
3	0.5447
4	0.4799

Fig. 4.8 shows the accuracy of the NASAFS feature set for the 12 class textile data with several noise realizations; feature set sizes range from 4 to 36 and are

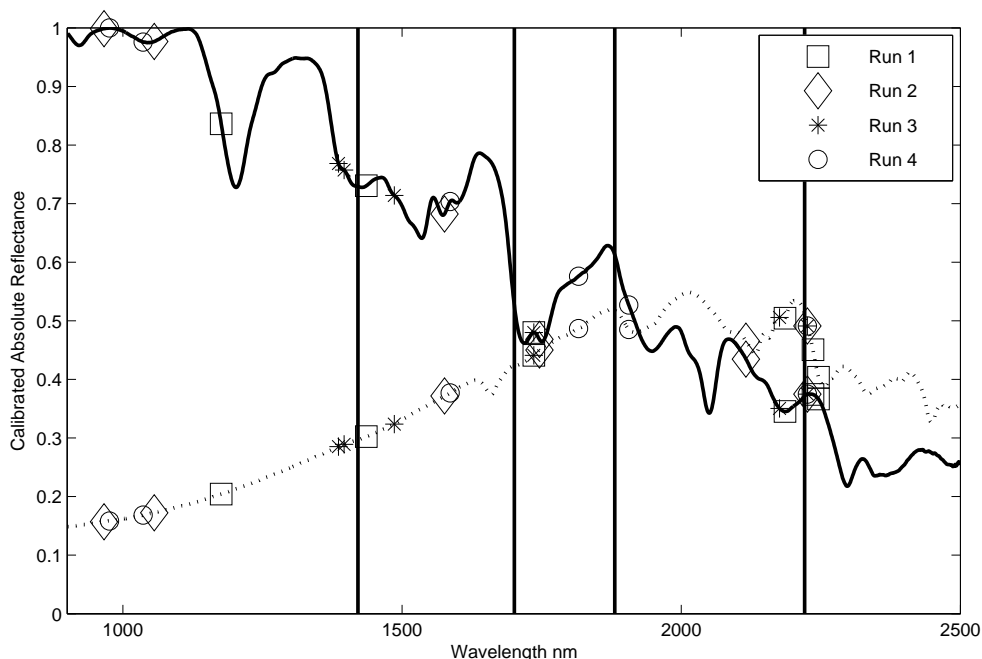


Figure 4.7: Reflectance spectra for 65% Polyester 35% Cotton blend (dashed) and 80% Nylon 20% Spandex blend (solid) signals with discriminating feature sets of four independent NASAFS runs: run 1 – square, run 2 – diamond, run 3 – asterisk, run 4 – circle.

evaluated in CoDeM. The accuracy increases as the feature set size increases, until a relative plateau occurs. For the 12 class textile data set, this plateau starts to occur at a feature set size of six for each noise realization; the selection of six features for the feature set size was partially driven by this observation. Likewise, NASAFS-IDF is used to evaluate the 12 class textile data set, at a 30% acceptable distribution and for each noise realization; several feature set sizes are used, ranging from 2 to 25 features, and are evaluated in CoDeM. The multi-feature set size NASAFS-IDF evaluation is shown in Fig. 4.9; in this instance, the accuracy appears to plateau at approximately four features for all noise realizations. The relationship between the feature set size and the correlation coefficient is shown in Fig. 4.10, again for a 30% acceptable distribution. Fig. 4.10 shows that as the feature set size increases, the correlation coefficient also increases. However, the correlation coefficient appears to

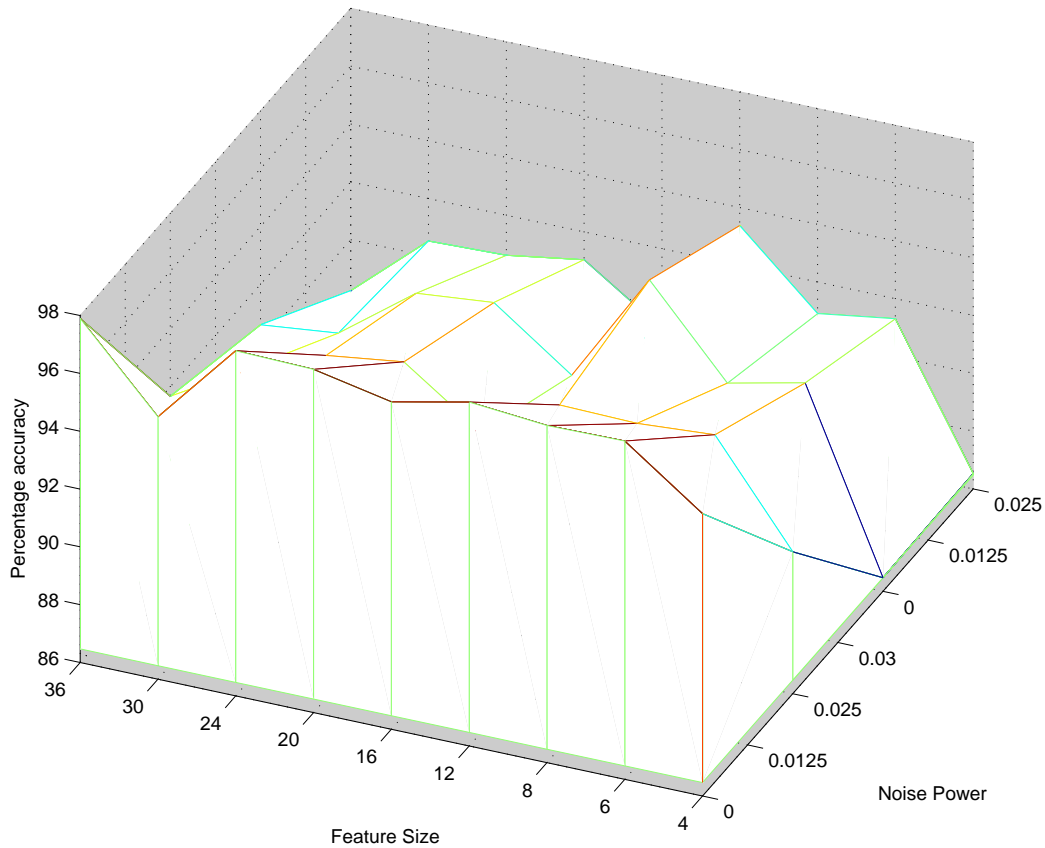


Figure 4.8: Accuracy of NASAFS, using CoDeM, of different sized feature sets (ranging from 4 to 36 features) over 4 noise realizations (from 0 to 0.03) for the 12 class textile data set.

decrease slightly after a feature set size of 15, but it effectively stabilizes at this point.

The 12 class textile data set is also evaluated by the ReliefF, GRLVQI, and Bhattacharyya feature selection methods. Each of these feature selection methods are evaluated using four different classification methods: CoDeM, Minimum Euclidean Distance (MED), Naïve Bayes, and C4.5. Table 4.4 shows the accuracy of each feature selection methodology using each of the aforementioned classification methods; these accuracies are compared to NASAFS and NASAFS-IDF (at 30% acceptable distributed spacing), using a bin size of 10 and a feature set size of six. For the

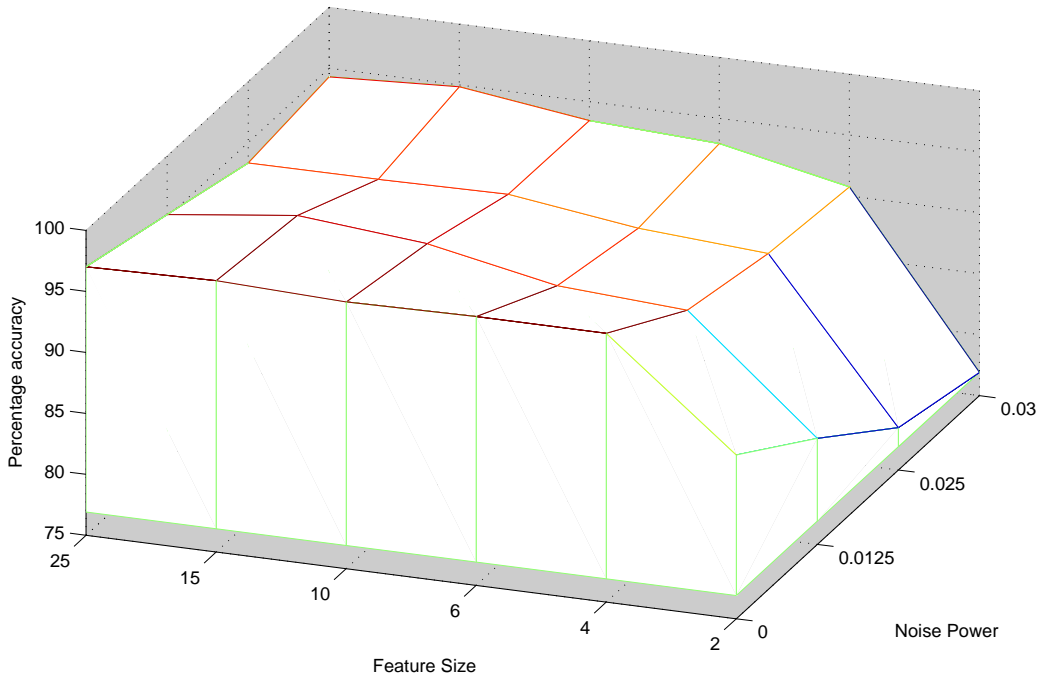


Figure 4.9: Accuracy of NASAFS-IDF, using CoDeM, of different sized feature sets (ranging from 2 to 25 features) over 4 noise realizations (from 0 to 0.03) for the 12 class textile data set.

12 class textile data set, although all of the feature set selection methods produced good results, For each classification method, NASAFS-IDF shows an improvement in accuracy over ReliefF, GRLVQI, and Bhattacharyya feature selection methods. The standard deviation as well as a significance test (discussed later) indicates that NASAFS-IDF holds a statistical significance over the other methods for each classifier; the one exception to this is ReliefF when using CoDeM. Table 4.4 shows the correlation coefficient for each of the feature sets; NASAFS-IDF is significantly less correlated than the other methods.

Fig. 4.11 shows the average accuracy of the feature selection methods, as classified by CoDeM, across a range of noise levels for the 12 class textile data set. As Fig. 4.11 shows, NASAFS-IDF classification results are better than those of the other methods. It is noted that as the noise level increases, the accuracies of NASAFS and NASAFS-IDF do not degrade as rapidly as do the accuracies of the other methods

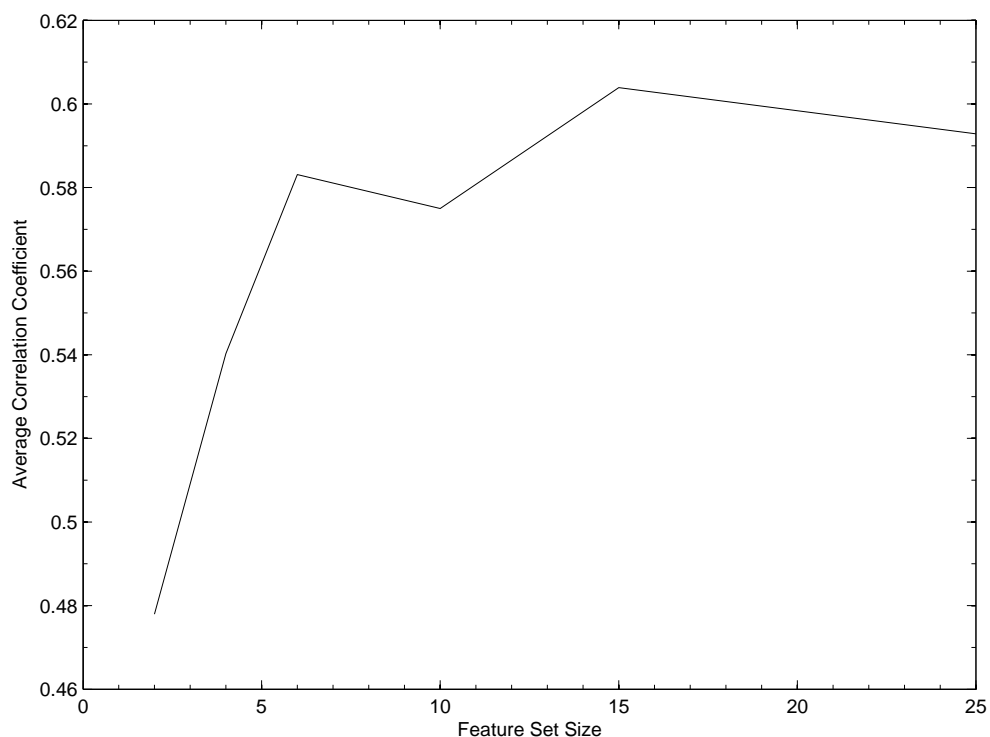


Figure 4.10: Diagram comparing average correlation coefficient of the feature set versus the feature set size for the 12 class textile data set. This is at 30% acceptable distribution for NASAFS-IDF.

evaluated. This provides for a more robust classification capability in the presence of noise.

The most notable difference in these methods is the correlation coefficient, which is an indication of the feature set's redundancy. The last column of Table 4.4 show each method's feature set correlation coefficient. NASAFS and NASAFS-IDF produce feature sets that have a correlation coefficient of 55% and 58%, respectively. The other methods compared produce feature sets with correlation coefficients in the 90% – 99% range.

Fig. 4.1 shows different class signals for the 12 class textile data set. It is evident from Fig. 4.1 that this data set contains correlated samples that create a difficult situation for classification, as well as impede the ability to produce non-redundant

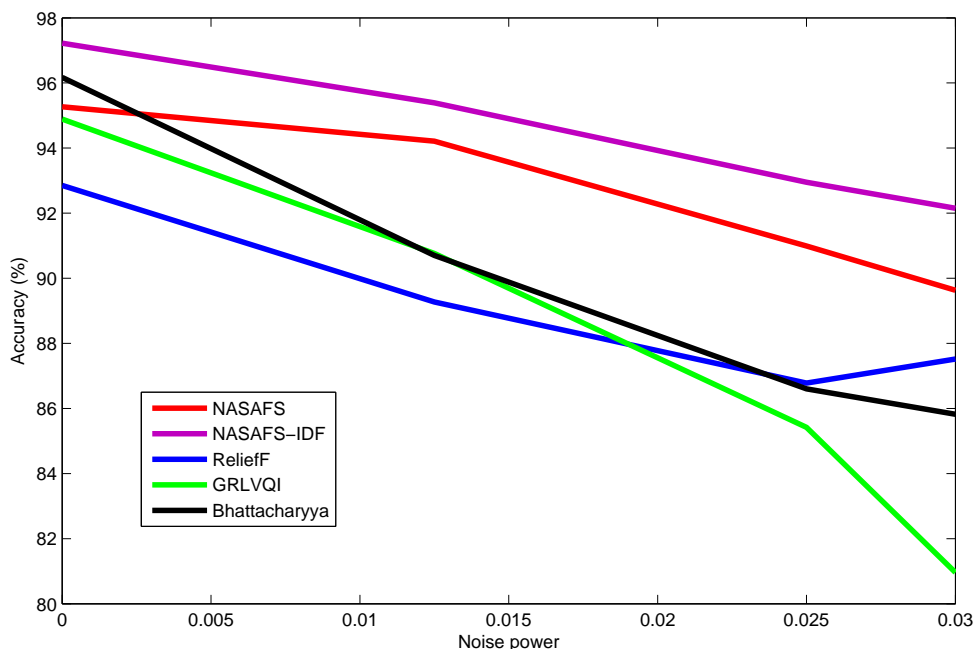


Figure 4.11: Results of the classification accuracy for the 12 class textile data set using CoDeM. Each feature selection method is represented with a different color: NASAFS (red), NASAFS-IDF (magenta), ReliefF (blue), GRLVQI (green), and Bhattacharyya (black).

feature sets with high classification accuracy. NASAFS and NASAFS-IDF have the ability to produce features that are distributed throughout the spectral domain; this is shown in Fig. 4.12. Fig. 4.12 shows the hyperspectral signal for 80% Polyester 20% Rayon Blend, along with the respective feature set elements for the previously mentioned feature selection methods. It is seen that the features selected by NASAFS and NASAFS-IDF are spread throughout the reference signal, whereas the features selected by the other methods tend to be clustered. In most cases, NASAFS and NASAFS-IDF also locate features of interest that correspond to areas that the other feature selection methods identified as producing good discriminating features.

A Pareto front for the objectives (accuracy and feature set size) is shown in Fig. 4.13 for the 12 class textile data set; this is for a 30% acceptable distributed spacing using CoDeM. In Fig. 4.13, the Pareto front is shown as circles with connected

Table 4.4: Accuracy and Average Correlation Coefficients for NASAFS, NASAFS-IDF, ReliefF, GRLVQI, and Bhattacharyya methods for the 12 class textile data set, where the feature size is six with no noise added to the data. The bin size for NASAFS-IDF is $10nm$ and the acceptable distributed spacing set to 30%.

Method	Classification/Detection					
	CoDeM	MED	Naïve Bayes	C4.5	μ, σ	\bar{r}
NASAFS	95.27%, $\pm na$	96.80%, $\pm na$	96.70%, $\pm na$	94.15%, $\pm na$	95.73%, $\pm na$	0.5469
NASAFS-IDF	96.64%, ± 0.01	98.36%, ± 0.02	97.69%, ± 0.25	97.67%, ± 0.38	97.59%, ± 0.45	0.5803
ReliefF	96.25%, ± 0.85	84.85%, ± 0.07	92.10%, ± 0.10	92.56%, ± 0.108	91.44%, ± 0.87	0.9955
GRLVQI	95.23%, ± 0.91	87.63%, ± 0.08	92.09%, ± 0.11	89.77%, ± 0.13	91.18%, ± 0.93	0.9025
Bhattacharyya	95.96%, ± 0.71	90.91%, ± 0.01	89.77%, ± 0.12	89.30%, ± 0.13	91.49%, ± 0.73	0.9933

lines, which indicate the non-dominated solutions for this data set. A contingency table is constructed for the 12 class textile data set for NASAFS-IDF; this is for a 30% acceptable distributed spacing with a zero noise level for the Naïve Bayes classifier. This contingency table is shown in Fig. 4.14, where the kappa statistic is noted as 0.9747 and the overall accuracy is 97.69%. The commission error, omission

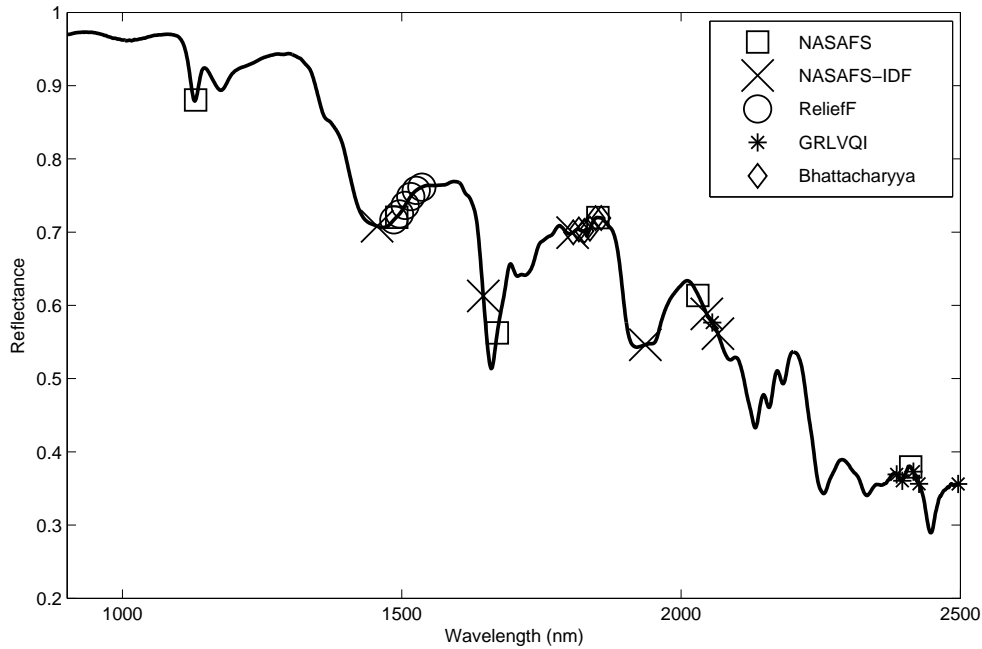


Figure 4.12: Hyperspectral signal for 80% Polyester 20% Rayon blend. The respective feature sets chosen by each feature selection method are indicated: NASAFS (box), NASAFS-IDF (X), ReliefF (circle), GRLVQI (asterisk), and Bhattacharyya (diamond).

error, producers accuracy, and consumers accuracy are displayed in Fig. 4.14, as well. NASAFS-IDF produces a feature set based on the one-versus-all concept; therefore, there are as many feature sets as the number of classes in the data set. The Naïve Bayes contingency table presents a challenge, since it uses a global feature set for classification; therefore, the row of the contingency table corresponding to the feature set being evaluated for Naïve Bayes is used in the contingency table shown in Fig. 4.14. From this new table, the errors and accuracies are computed; these are shown in Fig. 4.14, as well.

NASAFS and NASAFS-IDF do not have any provisions for a global feature set; however, a global set can be assumed by choosing the class feature set that displays the greatest overall accuracy. To provide an example, in the case of the textile data set

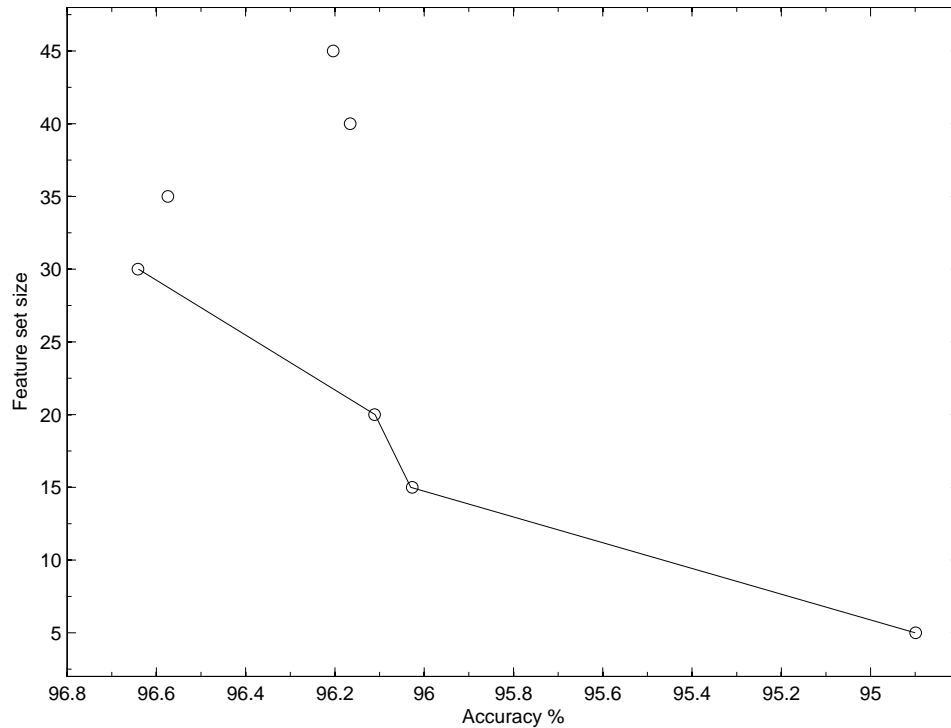


Figure 4.13: Pareto Front for the 12 class textile data set using CoDeM for the 30% acceptable distributed spacing criteria. The Pareto Front is for accuracy versus feature set size and is indicated by circles connected by lines.

using Naïve Bayes, each feature set is processed and its ability to accurately classify each sample to the appropriate class is recorded. In this particular instance, the feature set associated with class 7 performs best or is most accurate across all the classes. The feature set of this class (class 7) can then be assumed to perform as a global feature set because its overall accuracy is greater than the overall accuracy for any other class' feature set. The contingency table for class 7's feature set is shown in Fig. 4.15. The overall accuracy is 98.14% with a standard deviation of 0.056 and a kappa statistic of 0.9797. This could be a possible solution to a global feature set. However, determining a global feature set was not an agenda for this work and will require further investigation; therefore, the overall reported results will remain as indicated in Fig. 4.14.

4.3.2 LCVF Data Results. The correlation matrix used for NASAFS is shown in Fig. 4.16; horizontal black lines indicate the different sub-regions chosen by NASAFS. Only four regions were selected by NASAFS for the LCVF data set; these are based on the correlation matrix of the data. Therefore, as the data sets change, the number and sizes of sub-regions that NASAFS uses to determine the distributed spacing will change. The LCVF data set was collected in the field and contains noise from the field collection process; therefore, noise was not added to

Actual Class	Classified as Class:												Total # Samples	P.A.	O.E.	
	1	2	3	4	5	6	7	8	9	10	11	12				
1	18	0	0	0	0	0	0	0	0	0	0	0	0	18	100.00%	0.00%
2	0	18	0	0	0	0	0	0	0	0	0	0	0	18	100.00%	0.00%
3	0	0	18	0	0	0	0	0	0	0	0	0	0	18	100.00%	0.00%
4	0	0	0	18	0	0	0	0	0	0	0	0	0	18	100.00%	0.00%
5	0	0	0	0	18	0	0	0	0	0	0	0	0	18	100.00%	0.00%
6	0	0	0	0	0	14	0	4	0	0	0	0	0	18	77.78%	22.22%
7	0	0	0	0	0	0	18	0	0	0	0	0	0	18	100.00%	0.00%
8	0	0	0	0	0	1	0	17	0	0	0	0	0	18	94.44%	5.56%
9	0	0	0	0	0	0	0	0	18	0	0	0	0	18	100.00%	0.00%
10	0	0	0	0	0	0	0	0	0	18	0	0	0	18	100.00%	0.00%
11	0	0	0	0	0	0	0	0	0	0	18	0	0	18	100.00%	0.00%
12	0	0	0	0	0	0	0	0	0	0	0	18	0	18	100.00%	0.00%
Total	18	18	18	18	18	15	18	21	18	18	18	18	18			
C.A.	100.00%	100.00%	100.00%	100.00%	100.00%	93.33%	100.00%	80.95%	100.00%	100.00%	100.00%	100.00%	100.00%			
C.E.	0.00%	0.00%	0.00%	0.00%	0.00%	6.67%	0.00%	19.05%	0.00%	0.00%	0.00%	0.00%	0.00%			

Total Accuracy	97.69%
Kappa Statistic	0.9747

Figure 4.14: Contingency table for 12 class textile data set as reported by Naive Bayes, using the feature sets of NASAFS-IDF with a 30% acceptable distributed spacing. C.A. is the consumers accuracy, C.E. is the consumer error, P.A. is the producers accuracy and O.E. is the omission error.

Correctly Classified Instances	211	98.1395%
Incorrectly Classified Instances	4	1.8605%
Kappa statistic	0.9797	

Actual Class	Classified as Class											
	1	2	3	4	5	6	7	8	9	10	11	12
1	17	0	0	0	0	0	0	0	0	0	0	0
2	0	18	0	0	0	0	0	0	0	0	0	0
3	0	0	18	0	0	0	0	0	0	0	0	0
4	0	0	0	18	0	0	0	0	0	0	0	0
5	0	0	0	0	18	0	0	0	0	0	0	0
6	0	0	0	0	0	14	0	4	0	0	0	0
7	0	0	0	0	0	0	18	0	0	0	0	0
8	0	0	0	0	0	0	0	18	0	0	0	0
9	0	0	0	0	0	0	0	0	18	0	0	0
10	0	0	0	0	0	0	0	0	0	18	0	0
11	0	0	0	0	0	0	0	0	0	0	18	0
12	0	0	0	0	0	0	0	0	0	0	0	18

Figure 4.15: Contingency table for 12 class textile data set as reported by Naïve Bayes, using the feature set of class 7 of NASAFS-IDF with a 30% acceptable distributed spacing.

this data set for its evaluation by NASAFS, NASAFS-IDF or any other method. The LCVF 23 class data set is also evaluated by the ReliefF, GRLVQI, and Bhattacharyya feature selection methods. Each of these feature selection methods are evaluated by four different classification methods: CoDeM, Minimum Euclidean Distance (MED), Naïve Bayes, and C4.5. Table 4.5 compares ReliefF, GRLVQI, and Bhattacharyya feature selection methods to NASAFS and NASAFS-IDF for the LCVF data set with 45% acceptable distributed spacing for a bin size of 10 and a feature set size of six. Table 4.5 includes the standard deviation of each result to provide more insight into the statistical significance of the results. NASAFS-IDF shows an improvement in accuracy over ReliefF, GRLVQI, and Bhattacharyya feature selection methods for the MED, Naïve Bayes, and C4.5 classifiers. The standard deviation indicates that NASAFS-IDF has statistical significance over the other methods for those three classifiers. The correlation coefficients for the feature sets in Table 4.5 show that NASAFS-IDF is significantly less correlated than the other methods.

NASAFS and NASAFS-IDF show accuracies comparable to or better than, in terms of accuracy, the other methods when using feature sets with low correlation

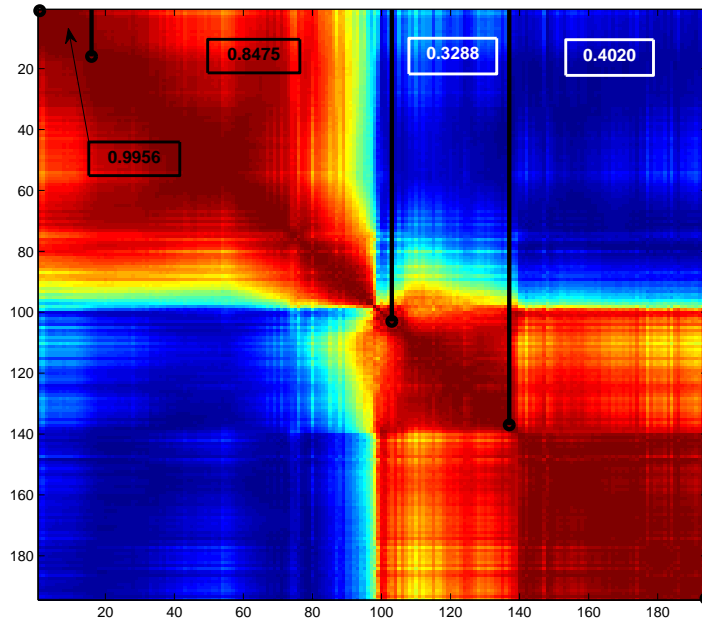


Figure 4.16: Correlation matrix of the 23 class LCVF data set. Sub-regions are determined by NASAFS and are marked with solid vertical black lines. The correlation coefficient of each sub-region is shown in the boxes.

coefficients (Fig. 4.5). NASAFS and NASAFS-IDF each produce a feature set that has a correlation coefficient of 40% and 48%, respectively. The other methods evaluated produce feature sets with correlation coefficients in the 98%–99% range; the exception is GRLVQI, which has a correlation coefficient of 71%.

The LCVF data set is a highly correlated data set (Fig. 4.2), which produces a difficult situation when attempting to select meaningful, non-redundant features for a feature set. From Fig. 4.5, it is evident that NASAFS and NASAFS-IDF can select highly discriminate features that have low correlation. Figure 4.17 shows a representative sample of the LCVF data set, with the feature locations shown as chosen by NASAFS, NASAFS-IDF, ReliefF, GRLVQI, and Bhattacharyya feature selection methods. It is evident, from Fig. 4.17, that while ReliefF and Bhattacharyya pick features that are clustered together (i.e redundant), GRLVQI attempts to pick

Table 4.5: Accuracy and Average Correlation Coefficients for NASAFS, NASAFS-IDF, ReliefF, GRLVQI, and Bhattacharyya methods for the LCVF data set, where the feature size is six with no noise added to the data. The bin size for NASAFS-IDF was $10nm$ and the acceptable distributed spacing is set to 45%.

<i>Method</i>	<i>Classification/Detection</i>					
	CoDeM	MED	Naïve Bayes	C4.5	μ, σ	\bar{r}
NASAFS	90.43%, $\pm na$	83.43%, $\pm na$	86.43%, $\pm na$	83.08%, $\pm na$	85.84%, $\pm na$	0.4027
NASAFS-IDF	89.02%, ± 0.68	85.84%, ± 0.07	89.57%, ± 0.41	85.81%, ± 0.53	87.56%, ± 0.96	0.4771
ReliefF	90.76%, ± 0.33	79.76%, ± 0.11	79.78%, ± 0.12	80.86%, ± 0.12	82.79%, ± 0.39	0.9961
GRLVQI	88.94%, ± 0.15	71.74%, ± 0.08	69.14%, ± 0.14	62.37%, ± 0.17	73.05%, ± 0.28	0.7058
Bhattacharyya	90.89%, ± 0.47	82.96%, ± 0.11	82.37%, ± 0.11	83.55%, ± 0.12	84.94%, ± 0.51	0.9836

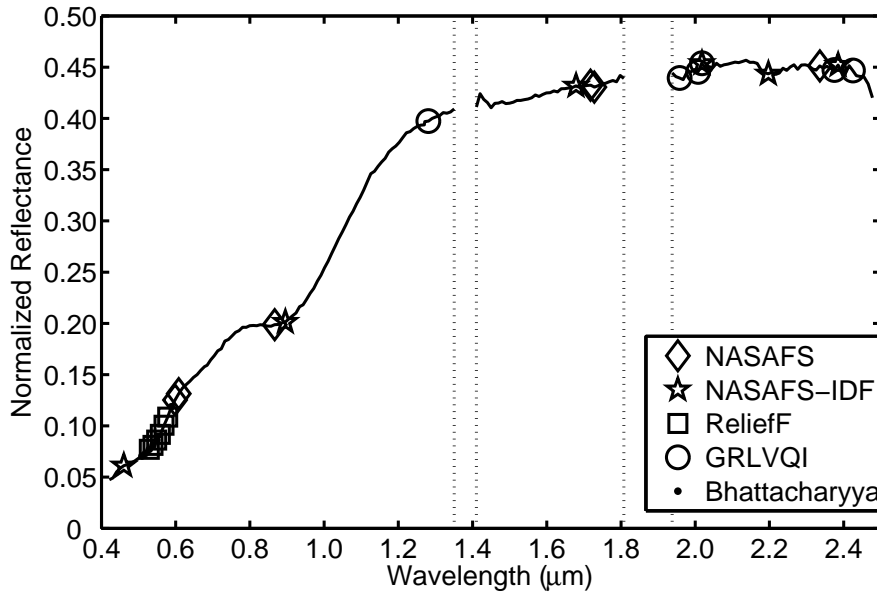


Figure 4.17: Representative sample of the LCVF 23 class data set, with the locations of the features of the feature sets selected by NASAFS (diamond), NASAFS-IDF (star), ReliefF (square), GRLVQI (circle), and Bhattacharyya (dot).

features that are less correlated; however, NASAFS and NASAFS-IDF select features that are more widely distributed throughout the sample domain.

A Pareto Front for the objectives (accuracy and feature set size) is shown in Fig. 4.18 for the LCVF 23 class data set; this is for a 45% acceptable distributed spacing using CoDeM. In Fig. 4.18, the Pareto front is shown as circles with connected lines; these indicate the non-dominated solutions for this data set.

A contingency table is constructed of the NASAFS-IDF results for the LCVF 23 class data set; these are calculated at a 45% acceptable distributed spacing and

a zero noise level for the Naïve Bayes classifier. This contingency table is shown in Fig. 4.19, where the kappa statistic is noted as 0.8868 and the overall accuracy is 89.57%. The commission error, omission error, producers accuracy, and consumers accuracy are displayed in Fig. 4.19. NASAFS-IDF produces a feature set based on the one-versus-all concept; therefore, there are as many feature sets as there are number of classes in a data set. As with the other data sets, producing a contingency table for Naïve Bayes presents a challenge; therefore, as before, the row of the contingency table corresponding to the feature set being evaluated for Naïve Bayes is used in the contingency table, as shown in Fig. 4.19. From this new table, the errors and accuracies are computed; these are shown in Fig. 4.19.

As previously explained, a global set can be determined by choosing the class feature set that displays the greatest overall accuracy. As described earlier, each feature set is processed by Naïve Bayes and its ability to accurately label each sample to the appropriate class is recorded. In this case, the feature set associated with class

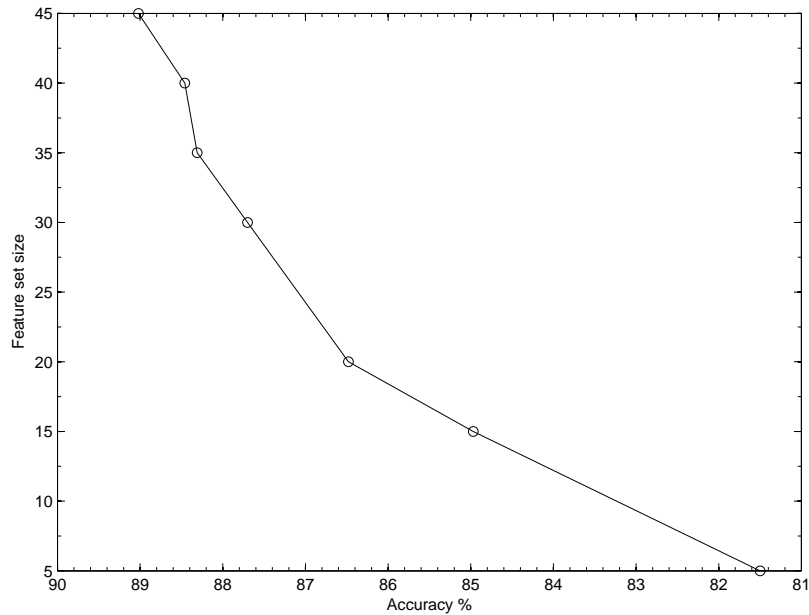


Figure 4.18: Pareto Front for the LCVF 23 class data set using CoDeM for the 45% acceptable distributed spacing criteria. The Pareto Front is accuracy versus feature set size and is indicated by circles connected by lines.

1 performs the best across all classes. Therefore, the feature set of this class can be assumed to perform as a global feature set because its overall accuracy is greater than the overall accuracy for any other class' feature set. The contingency table for class 1's feature set is shown in Fig. 4.20. The overall accuracy is 93.66% with a standard deviation of 0.07 and a kappa statistic of 0.9312. As with the textile data results, these statistics display greater accuracy than the average statistics previously reported. However, the overall reported results will remain as indicated in Fig. 4.19; this is due to the fact that a global feature set was not an original goal of this work, and further study will be necessary before conclusions can be drawn.

4.3.3 Texture Data Results. The results of the 5 level wavelet decomposition of the 7 class texture data set are processed with NASAFS-IDF, ReliefF, GRLVQI, and Bhattacharyya only; therefore, there are no performance comparisons of NASAFS-IDF to NASAFS for this data set. NASAFS is not evaluated with this data set due to the structure of the correlation matrix. NASAFS must be able to determine sub-regions (regions of strong correlation) of the data. This data set does not present an adequate region that could be divided for the use of sub-regions. This can be visually identified in Fig. 4.21 which shows the correlation matrix of the texture data set. The

Actual Class	Classified as Class:																							Total # Samples	P.A.	O.E.	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23				
1	70	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	71	98.59%	1.41%	
2	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	81.82%	0.00%	
3	0	0	37	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	4	0	0	50	74.00%	18.00%	
4	0	0	0	157	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	160	98.13%	1.88%	
5	0	0	0	0	115	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	115	100.00%	0.00%	
6	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	100.00%	0.00%	
7	0	0	0	0	0	1	5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	7	71.43%	14.29%	
8	0	0	0	0	0	0	0	49	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	50	98.00%	0.00%	
9	0	0	5	0	0	0	0	2	21	0	0	0	2	0	0	1	0	0	0	0	1	0	4	36	58.33%	19.44%	
10	0	0	0	0	0	0	0	0	11	0	0	1	0	0	0	0	0	0	0	0	0	0	0	12	91.67%	0.00%	
11	0	0	0	0	0	0	0	0	0	35	0	0	0	0	0	1	0	0	0	0	0	1	0	37	94.59%	0.00%	
12	0	0	0	0	0	0	0	0	0	0	0	73	0	0	0	0	0	0	0	0	0	0	0	5	78	93.59%	0.00%
13	0	0	0	0	0	0	0	2	2	0	0	0	10	0	0	0	0	0	0	0	0	0	0	14	71.43%	28.57%	
14	0	0	0	0	0	0	0	2	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	15	86.67%	13.33%	
15	0	0	0	0	0	0	0	0	1	0	0	0	0	0	52	1	0	0	0	0	0	0	0	54	96.29%	3.70%	
16	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	41	0	0	0	0	0	0	0	45	91.11%	8.89%	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	15	100.00%	0.00%	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	5	0	0	0	0	14	64.29%	35.71%	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	15	100.00%	0.00%	
20	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	14	0	0	0	18	77.78%	22.22%	
21	0	0	4	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	29	0	0	36	80.56%	19.44%	
22	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	2	0	0	0	0	0	7	0	12	58.33%	41.67%	
23	0	0	3	0	0	0	0	2	0	0	0	7	3	0	0	0	0	0	0	0	2	0	0	16	33	48.48%	51.52%
Total	70	18	49	157	119	23	5	55	37	11	41	81	20	13	55	47	15	9	21	14	37	8	25				
C.A.	100.00%	100.00%	75.51%	100.00%	96.64%	91.30%	100.00%	89.09%	56.76%	100.00%	85.37%	90.12%	50.00%	100.00%	94.55%	87.23%	100.00%	100.00%	71.43%	100.00%	78.38%	87.50%	64.00%				
C.E.	0.00%	0.00%	24.49%	0.00%	3.36%	8.70%	0.00%	10.91%	43.24%	0.00%	14.63%	9.88%	50.00%	0.00%	5.45%	12.77%	0.00%	0.00%	28.57%	0.00%	21.62%	12.50%	36.00%				

Total Accuracy	89.57%
Kappa Statistic	0.8868

Figure 4.19: Contingency table for LCVF 23 class data set as reported by Naïve Bayes, using the feature sets of NASAFS-IDF with a 45% acceptable distributed spacing. C.A. is the consumers accuracy, C.E. is the consumer error, P.A. is the producers accuracy and O.E. is the omission error.

texture data set is processed, over four noise realizations and multiple feature set sizes, to show the performance based on a feature set size for 35% acceptable distributed spacing. Figure 4.22 shows that for feature sizes up to 25, it peaks at 15; however, the difference in accuracy between a feature size of six and a feature size of 15 is small enough to justify using a feature size of six to process the remainder of this data set. The correlation between the feature set size and the correlation coefficient, at a 35% acceptable distribution, is shown in Fig. 4.23. This figure shows that, in general, as the feature set size increases, the correlation coefficient also increases. However, it is noted that for a feature set size of two the correlation coefficient is higher than the correlation coefficient for a feature set size of four. After feature set size reaches 15,

Correctly Classified Instances	871	93.66%
Incorrectly Classified Instances	59	6.34%
Kappa statistic	0.931	

Actual Class	Classified as Class																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	70	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
2	0	15	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	4
3	0	0	46	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	1	
4	0	0	0	159	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	115	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	11	0	0	1	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	36	0	0	0	0	1	0	0	0	0	0	0	0
12	0	2	0	0	0	0	0	0	0	0	0	73	0	0	0	0	0	0	0	0	0	0	3
13	0	0	1	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	2	0	0	0	0	7	6	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	1	0	0	0	0	0	51	2	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	4	0	0	0	1	40	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	3	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	11	1	0	0	0	0
20	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0
21	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	34	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	10	0	0
23	0	4	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	24

Figure 4.20: Contingency table for 23 class LCVF data set as reported by Naïve Bayes, using the feature set of class 1 of NASA FS-IDF with a 45% acceptable distributed spacing.

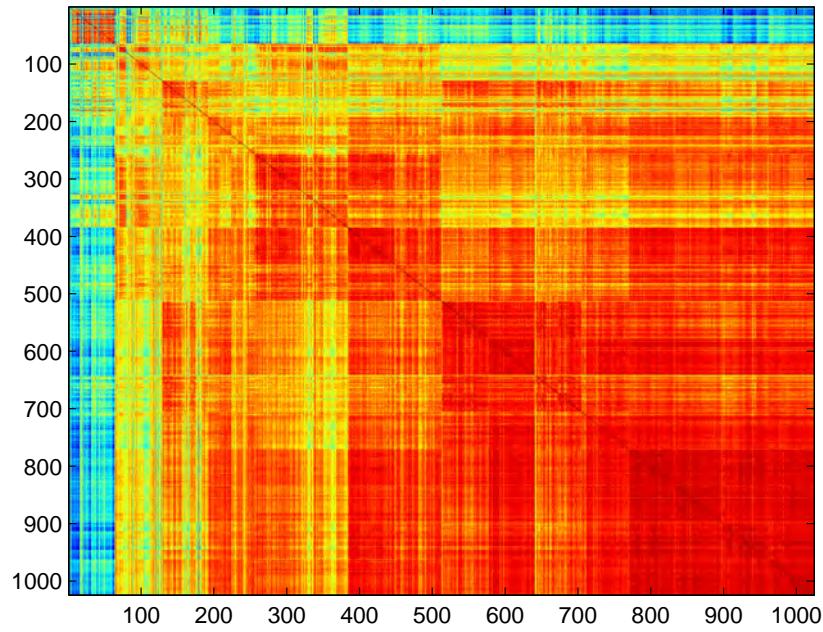


Figure 4.21: Correlation matrix of the 7 class texture data set.

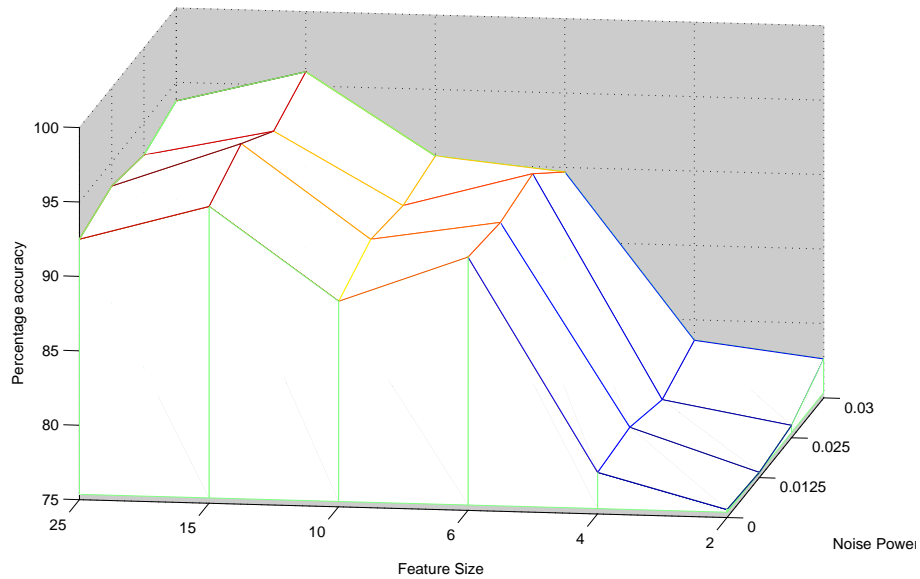


Figure 4.22: Accuracy of NASAFS-IDF, using CoDeM, of different sized feature sets (ranging from 2 to 25 features) over 4 noise realizations (from 0 to 0.03) for the 7 class texture data set for a 35% acceptable distribution.

the correlation coefficient appears to slightly decrease; however, it is predominantly stabilized at this point for the feature set sizes evaluated. This unexpected behavior of the correlation coefficient correspondence to the feature set size is due to the fact that this data set is not strictly correlated. Figure 4.24 is a representative sample of the texture data set. The data set is preprocessed via the wavelet decomposition, resulting in the values being highly uncorrelated between branches. This fact could lead to the results shown in Fig. 4.23.

Table 4.6 compares ReliefF, GRLVQI, and Bhattacharyya feature selection methods to NASAFS-IDF for the texture data set, at an acceptable distributed spacing of 35%, a bin size of 10, and a feature set size of six. Table 4.6 includes the standard deviations of each result in order to provide more information regarding the statistical significance of the results. NASAFS-IDF shows an improvement in accuracy over ReliefF, and Bhattacharyya feature selection methods for CoDeM, Naïve Bayes, and C4.5 classification methods. However, the standard deviation indicates that NASAFS-IDF

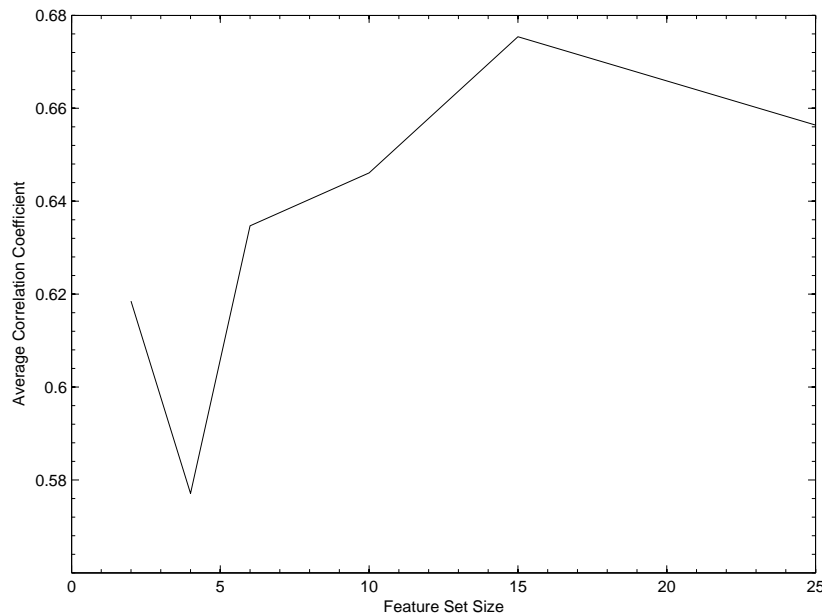


Figure 4.23: Diagram comparing the average correlation coefficient of the feature set versus the feature set size for the 7 class texture data set at 35% acceptable distribution for NASAFS-IDF.

only displays a statistical significance over the other methods for Naïve Bayes and C4.5 classifiers. The last column of Table 4.6 shows the correlation coefficient for the feature sets and verifies that NASAFS-IDF is less correlated than the other methods. The correlation coefficient is the most important difference between these methods,

Table 4.6: Accuracy and Average Correlation Coefficients for NASAFS-IDF, ReliefF, GRLVQI, and Bhattacharyya methods for the 7 class texture data set, where the feature size is six with no noise added to the data. The bin size for NASAFS-IDF was $10nm$ and the acceptable distributed spacing is set to 35%.

<i>Method</i>	<i>Classification/Detection</i>					
	CoDeM	MED	Naïve Bayes	C4.5	μ, σ	\bar{r}
NASAFS-IDF	87.72%, ± 2.82	80.95%, ± 0.10	72.29%, ± 0.65	63.86%, ± 0.82	76.21%, ± 3.01	0.6598
ReliefF	87.50%, ± 1.97	86.81%, ± 0.09	69.88%, ± 0.25	61.45%, ± 0.29	76.41%, ± 2.01	0.7507
GRLVQI	87.24%, ± 3.71	91.67%, ± 0.04	79.52%, ± 0.21	63.86%, ± 0.31	80.57%, ± 3.73	0.6724
Bhattacharyya	86.11%, ± 1.84	83.33%, ± 0.09	71.08%, ± 0.25	54.22%, ± 0.34	73.69%, ± 1.89	0.8381

because it is an indication of feature set redundancy. However, due to the nature of this data set, the redundancy of features might not matter. Fig. 4.25 illustrates the spread of the features for each feature selection method’s feature set as plotted against a class 1 and 2 texture sample. It is noted from Fig. 4.25 and Table 4.6 that feature spreading has little to do with accuracy for this data set and noise level. Fig. 4.26 is

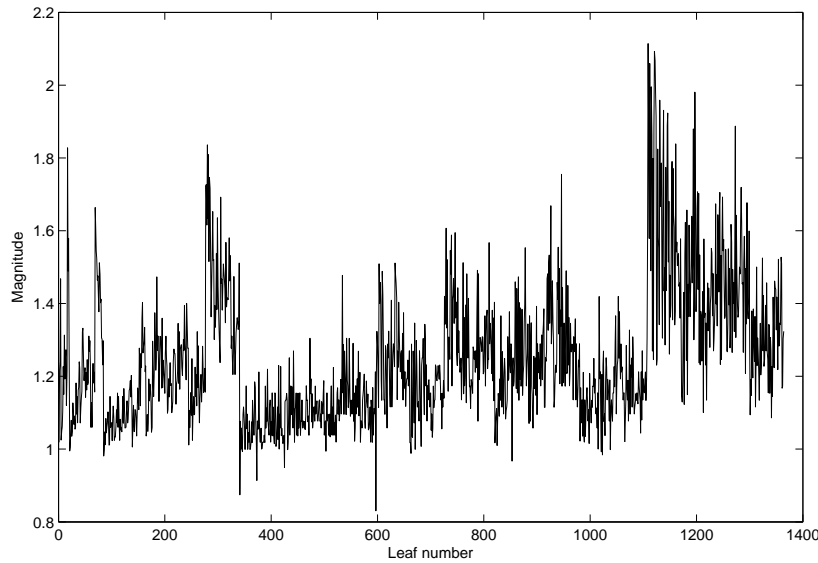


Figure 4.24: Selected representative sample for the 7 class texture data set.

an exert from Fig. 4.25 to show more detail of the two classes. The Bhattacharyya method is more clustered than the others and has comparable accuracies. It can also be noted that due to the nature of this data set that all of the feature selection methods chose features that are spread across the domain. NASAFS-IDF produces a feature set that has a correlation coefficient of 0.66, whereas the correlation coefficients obtained by the other methods are higher, but not extremely so. Again, this is attributed to the fact that the data in this data set is not strictly correlated. A Pareto front for the objectives (accuracy and feature set size) of the 7 class texture data set is shown in Fig. 4.27; these are for a 35% acceptable distributed spacing using CoDeM. In Fig. 4.27, the Pareto front is shown as circles with connected lines; this indicates the non-dominated solutions for this data set. A contingency table is constructed for the 7 class texture data set NASAFS-IDF results; this table displays a 35% acceptable distributed spacing with a zero noise level for the Naïve Bayes classifier. It is shown in Fig. 4.28, where the kappa statistic is noted as 0.68 and the overall accuracy is 72.29%.

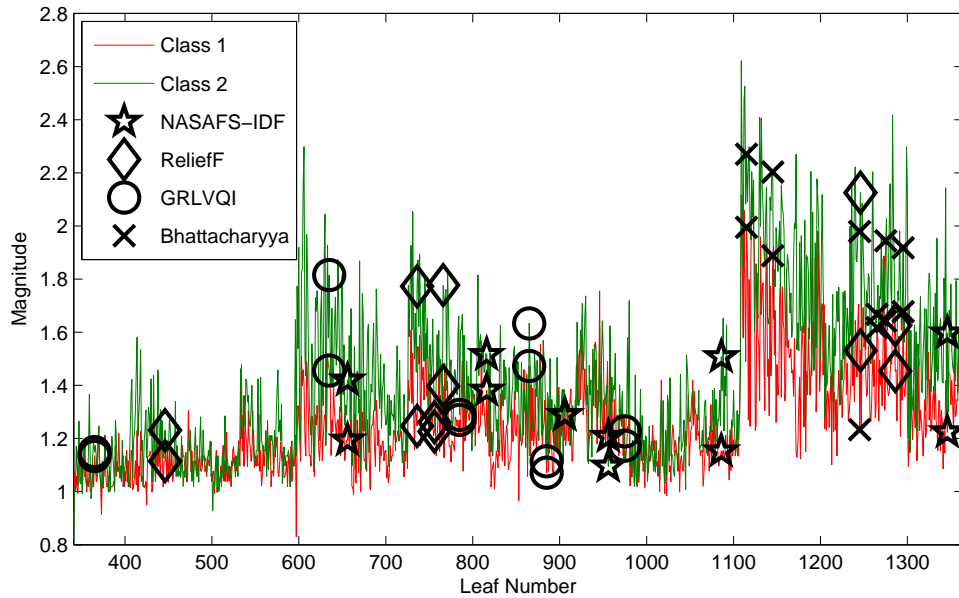


Figure 4.25: Class 1 and 2 sample of the 7 class texture data set. The features of each feature selection method are indicated as follows: NASAFS-IDF star, ReliefF diamond, GRLVQI circle, and Bhattacharyya 'x'.

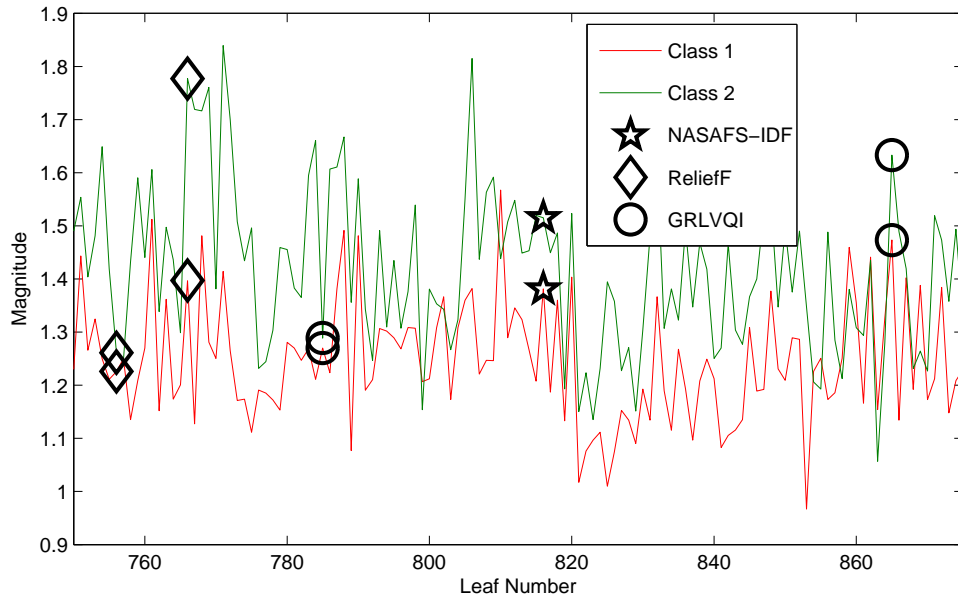


Figure 4.26: Class 1 and 2 sample of the 7 class texture data set. The features of each feature selection method are indicated as follows: NASAFS-IDF star, ReliefF diamond, and GRLVQI circle.

The commission error, omission error, producers accuracy, and consumers accuracy are displayed in Fig. 4.28, as well. As previously stated, NASAFS-IDF produces a feature set based on the one-versus-all concept; as such, the number of feature sets is equal to the number of classes in a data set. A contingency table is produced for Naïve Bayes. The row of the contingency table corresponding to the feature set being evaluated for Naïve Bayes is used in the contingency table shown in Fig. 4.28. From this new table, the errors and accuracies are computed; these are shown in Fig. 4.28, as well. Each feature set is processed by Naïve Bayes in an attempt to determine a global feature set for the data. In this data set, each sample is labeled and the results are recorded. The feature set associated with class 2 outperforms the feature sets for all other classes. The class 2 feature set contingency table is shown in Fig. 4.29. The overall accuracy is 86.75%, with a standard deviation of 0.18 and a kappa statistic of 0.85. As with other data sets, the statistics obtained using class 2 as a global feature set show more desirable results than the previously reported statistics. However, as

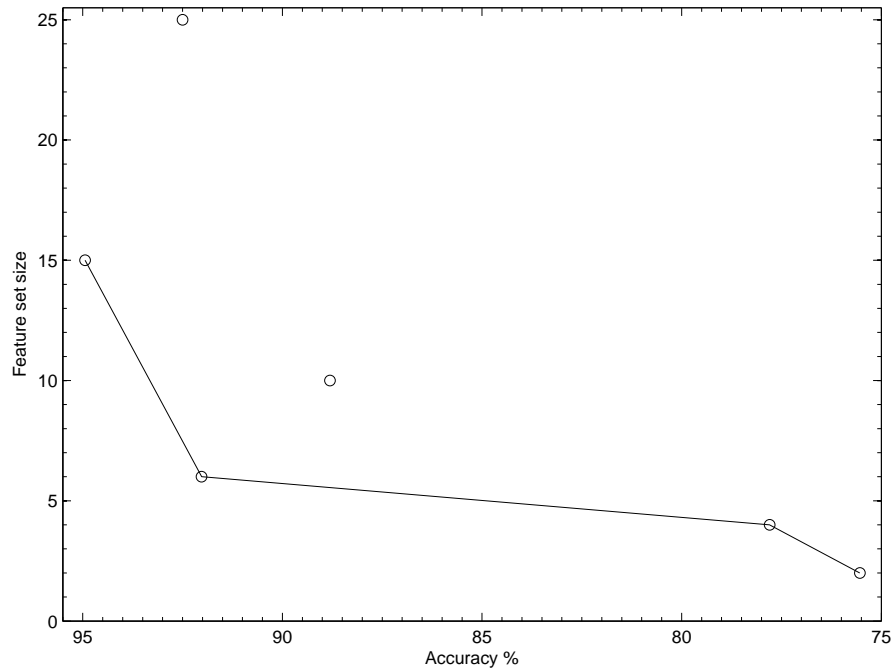


Figure 4.27: Pareto Front for the 7 class texture data set using CoDeM for the 35% acceptable distributed spacing criteria. The Pareto Front is accuracy versus feature set size and is indicated as circles connected by lines.

Actual Class	Classified as Class:							Total # Samples	P.A.	O.E.
	1	2	3	4	5	6	7			
1	9	0	0	0	0	0	2	11	81.82%	18.18%
2	0	10	1	0	0	0	1	12	83.33%	16.67%
3	1	4	1	1	0	0	5	12	8.33%	91.67%
4	0	0	2	10	0	0	0	12	83.33%	16.67%
5	0	0	0	0	12	0	0	12	100.00%	0.00%
6	0	0	0	0	0	12	0	12	100.00%	0.00%
7	2	3	0	1	0	0	6	12	50.00%	50.00%
Total	12	17	4	12	12	12	14			
C.A.	75.00%	58.82%	25.00%	83.33%	100.00%	100.00%	42.86%			
C.E.	25.00%	41.18%	75.00%	16.67%	0.00%	0.00%	57.14%			

Total Accuracy	72.29%
Kappa Statistic	0.6767

Figure 4.28: Contingency table for 7 class texture data set as reported by Naïve Bayes, using the feature sets of NASAFS-IDF with a 35% acceptable distributed spacing. C.A. is the consumers accuracy, C.E. is the consumer error, P.A. is the producers accuracy and O.E. is the omission error.

has been stated, determining a global feature set was not an agenda of this work. Further, more thorough testing is required; therefore, the results reported in Fig. 4.28 will remain as indicated.

4.4 Significance of Feature Selection Methods

For each data set, NASAFS and NASAFS-IDF (where appropriate) produced better results that have lower correlation among the features of the feature set. The computation cost required by NASAFS-IDF is less than is required for NASAFS, and NASAFS-IDF appears to outperform NASAFS in some instances. The statistical significance of NASAFS-IDF, as compared to the other feature selection methods Tables 4.4, 4.5, 4.6, is due in part to the standard deviation. To further test the significance of NASAFS-IDF against the other feature selection methods tested for all three data sets, the Wilcoxon signed-rank test is applied. NASAFS is not used with the 7 class texture data set and therefore the Wilcoxon signed-rank test is only computed for all five feature selection techniques for only the hyperspectral data set results.

The Wilcoxon signed-rank test is a non-parametric test that does not assume normality of the data. It produces a test statistic that is then compared to a chart

Correctly Classified Instances	72	86.75%
Incorrectly Classified Instances	11	13.25%
Kappa statistic	0.8454	

Actual Class	Classified As						
	1	2	3	4	5	6	7
1	11	0	0	0	0	0	0
2	0	10	1	0	0	0	1
3	0	2	9	0	0	0	1
4	0	0	0	12	0	0	0
5	0	1	0	0	11	0	0
6	0	0	0	0	0	12	0
7	2	2	0	1	0	0	7

Figure 4.29: Contingency table for 7 class texture data set as reported by Naïve Bayes, using the feature set of class 2 of NASAFS-IDF with a 35% acceptable distributed spacing.

of critical values, in order to determine the significance of a random variable or the significance of one random variable versus another [92], [82], [53], [81]. In order to apply this test to the data sets that have been used for our evaluations, we must first define the data sets being tested.

We will compare the feature selection methods to each other by assembling the accuracy results into a vector. Each feature selection method (i.e. NASAIFS-IDF, ReliefF, GRLVQI, and Bhattacharyya) will have a vector of values associated with each accuracy, using each classifier (i.e. MED, CoDeM, Naïve Bayes, and C4.5) for each set of data (i.e. 12 class textile, 23 class LCVF, 7 class texture), resulting in each vector having 12 values. This data set of the feature selection methods' accuracies must be checked for normality. If the data is determined to be normally distributed, another type of test (e.g. paired t-test) must be used. For the purposes of our work, the Shapiro Wilks test will be used to test for normality.

The Shapiro Wilks test determines if a sample came from a normal distribution; it is a good testing choice because it works well with small sample sizes. The test statistic for this test is compared to a table in order to determine rejection or non-rejection of the null hypothesis of normality [27, 80]. If the test statistic falls below the 50% value of Table 6 in [80], then the null hypothesis is rejected and the sample is considered to be from a non-normal distribution. As previously stated, we have 12 values per vector; therefore, according to 'the analysis of variance test for normality' table (Table 6) of [80], since $n = 12$, the 50% value is 0.943. The test statistic for NASAIFS-IDF is 0.864, ReliefF is 0.920, GRLVQI is 0.868, and Bhattacharyya is 0.810. All fall below the 50% value of 0.943 and are therefore considered to be from a non-normal distribution, and we can proceed with the Wilcoxon signed-rank test.

The Wilcoxon signed-rank test is typically used on data that cannot meet the requirements of a t-test; namely, that the scale of the values for each variable be of an equal interval scale, and that the variables belong to a normal distribution [53]. Our data does not meet either of these t-test criteria. As determined by the Shapiro

Wilks test, the data is not from a normal distribution. Additionally, even though the values are given as a percentage from 0 to 100, they are determined by different methods, meaning the scale of the values in each variable can be argued to not be of an equal interval scale. Therefore, the Wilcoxon signed-rank test will be a good fit for our data.

The basic operation of the Wilcoxon signed-rank test is to first define a null hypothesis. The null hypothesis is typically set in order to determine a specific value of the two vectors; the alternative hypothesis is that which is not the null (i.e. $H_0 = 4$ and H_a otherwise). The definitions set by the null and alternative hypothesis determine whether the critical value table to be used should be from a one-tailed or two-tailed test. This is dependent on directionality. If the null hypothesis is set to be a value or less and the alternative is to be larger than that value, then the critical values to be used come from a one-tailed test [81]. If the null hypothesis is set to be a value, and the alternative hypothesis is set to be any value that is not that value, then the critical values to be used come from a two-tailed test. For our situation, the null hypothesis is *there is no significant difference in method A versus method B*, and the alternative is *method A is significantly better than method B*; therefore, we use a one-tailed test. Table 4.7 shows the results of the Wilcoxon signed-rank test for the case using the results from all three data sets, where the row corresponds to method A and the column corresponds to method B. For the result located in row 1 column 3, the chart reads in the following manner: NASAFS-IDF method outperforms the ReliefF method, with a 0.025 significance. It can be seen that for the case where NASAFS-IDF is compared against the other methods, NASAFS-IDF is consistently better, by a significance of 0.025 or greater. When the other methods are tested against NASAFS-IDF or each other, the null hypothesis is not rejected; therefore, it is determined that the other methods tested against NASAFS-IDF are not better to any significance level. Table 4.8 shows the results of the Wilcoxon signed-rank test for the case using the results from the hyperspectral data sets only. It can be seen that NASAFS is able to reject the null hypothesis when compared against ReliefF

and GRLVQI to a significance level of 0.025 and 0.01 respectively. NASAFS-IDF rejected the null hypothesis when tested against all other feature selection methods even NASAFS. Table 4.8 shows a level of significance over NASAFS to a significance level of 0.025 and 0.01 for the other methods tested.

Table 4.7: Wilcoxon signed-rank test results for the feature selection methods tested in this work. The row is considered as method *A* and the column is considered as method *B*. For the Wilcoxon signed-rank test, and as our null hypothesis states for this test, we are determining if method *A* is better than method *B* with a significance better than 0.05.

	NASAFS-IDF	ReliefF	GRLVQI	Bhattacharyya
NASAFS-IDF	—————	0.025	0.025	0.01
ReliefF	Not Rejected	—————	Not Rejected	Not Rejected
GRLVQI	Not Rejected	Not Rejected	—————	Not Rejected
Bhattacharyya	Not Rejected	Not Rejected	Not Rejected	—————

Table 4.8: Wilcoxon signed-rank test results for all five feature selection methods tested in this work. The row is considered as method *A* and the column is considered as method *B*. For the Wilcoxon signed-rank test, and as our null hypothesis states for this test, we are determining if method *A* is better than method *B* with a significance better than 0.05.

	NASAFS	NASAFS-IDF	ReliefF	GRLVQI	Bhattacharyya
NASAFS	—————	Not Rejected	0.025	0.01	Not Rejected
NASAFS-IDF	0.025	—————	0.01	0.01	0.01
ReliefF	Not Rejected	Not Rejected	—————	0.025	Not Rejected
GRLVQI	Not Rejected	Not Rejected	Not Rejected	—————	Not Rejected
Bhattacharyya	Not Rejected	Not Rejected	Not Rejected	0.05	—————

4.5 Summary

Three different data sets are evaluated by our novel feature selection methods: 12 class textile data set, Lunar Crater Volcanic Field data set (LCVF), and 7 class Brodatz texture data set. The results of these evaluations are compared to three feature selection methods: ReliefF, GRLVQI, and Bhattacharyya. For each data set, each of the feature selection methods is then classified by four classification methods: CoDeM, MED, Naïve Bayes, and C4.5. The results for each data set are shown in Sections 4.3.1, 4.3.2, and 4.3.3. On the average, our novel feature selection methods (NASAFS and NASAFS-IDF) outperform all the other feature selection methods

evaluated for all the data sets, and show statistical significance according to the standard deviation of each test. This performance is consistent with all the classifiers used; however, for the hyperspectral data sets, Naïve Bayes performs best for our novel feature selection methods. Naïve Bayes uses the basic assumption of relative independence; the low correlated feature sets produced by NASAFS and NASAFS-IDF statistically resemble relative independence. This could explain why Naïve Bayes produces better results than the other classifiers for our feature selection method. The correlation coefficient for each feature selection method is seen in Tables 4.4, 4.5, and 4.6; these values show that the correlation coefficients produced by NASAFS and NASAFS-IDF are significantly lower than the correlation coefficients produced by the other feature selection methods evaluated.

The Pareto front of each NASAFS-IDF data set is shown in Fig. 4.13, 4.18, and 4.27, where the effect of feature set size versus accuracy can be determined. As expected, larger feature set size achieves better accuracy; however, this is only to a point, after which the proliferation of features overwhelms the classifiers and begins to decrease classification accuracy. The exception to this finding, which is the LCVF data set, did not show a decrease in accuracy as chosen feature set size increases; however, the rate that the accuracy increases decreased. It is also determined that as the feature set size increases, the correlation coefficient also increases (Fig. 4.10, and 4.23); this is as would be expected.

A contingency table is created from Naïve Bayes results for NASAFS-IDF (Fig. 4.14, 4.19, and 4.28), and for each of the data sets shown, the kappa statistic is extremely high; this provides additional evidence that NASAFS-IDF has statistical relevance and exceptional accuracy as a feature selection methodology. It should also be noted that the texture data set is not a hyperspectral data set, and it is not continuously correlated as in the manner of hyperspectral data sets. Our intent is for our novel feature selection systems to be used on hyperspectral data sets; however, this data set is evaluated to determine the flexibility of our feature selection methodology when used with other types of data sets. Upon evaluation, it is found

that NASAFS-IDF produces results that fall well within the range of acceptability. Finally, the results of the Wilcoxon signed-rank test are shown in Table 4.7 and 4.8. These results show that NASAFS-IDF is superior to other feature selection methods, when compared over all data sets for all classifiers, to a significance of 0.025 to 0.01.

V. Conclusion

The ability to locate an object in any the environment has a multitude of applications. Specifically, locating a dismount in the scene is useful for Search and Rescue, Military operations, and Security purposes. This objective can be aided with an accurate textile detection method. The ability to accurately detect textiles depends on distinguishing between different material types. Hyperspectral data provides a multitude of distinguishing characteristics to aid in detection.

However, to use hyperspectral data in a computationally efficient way, one must pair-down the information space to only a few key bits of highly discriminating features. Feature selection is a method that accomplishes this capability. However, not all feature selection methods select highly discriminatory features that are non-redundant. Non-redundancy provides robustness of classification, especially in the presence of noise.

The goal posed at the beginning of this work is to develop a feature selection methodology that randomly and non-greedily chooses features to achieve a minimally-sized, non-redundant feature set that produces accurate classification. NASAFS and NASAFS-IDF are designed to be used on continuous highly dimensional data sets; however, various methodologies, taxonomies, and processes that encompass common feature selection methodologies are researched to explore and demonstrate capability with other types of data sets.

NASAFS and NASAFS-IDF use a stochastic search algorithm in conjunction with a heuristic that combines measures of distance and dependence to select features. They also incorporate a distributed spacing equation to produce low-correlated feature sets. NASAFS and NASAFS-IDF are able to produce results that are highly accurate. These accuracies out perform those of the other methods tested. NASAFS and NASAFS-IDF also produce feature sets with a correlation coefficient that are about half that of the other methods tested.

5.1 *Summary of Results*

This dissertation presented the novel NASAFS and NASAFS-IDF feature selection methods that produce small feature sets which achieve non-redundancy and excellent classification accuracy. NASAFS and NASAFS-IDF, differ in their method of determining highly discriminate features that have low correlation. However, both achieve excellent results.

Three common feature selection methods are compared to NASAFS and NASAFS-IDF; ReliefF, GRLVQI, and Bhattacharyya (Section 2.2). Three different data sets are used to validate the capability of NASAFS and NASAFS-IDF, two hyperspectral data sets, one lab quality and the other field collected, and a non-correlated (Section 4.1.1). NASAFS and NASAFS-IDF consistently choose low correlated feature sets for all three of the data sets tested (Tables 4.4, 4.5, 4.6). NASAFS and NASAFS-IDF typically outperform ReliefF, GRLVQI, and Bhattacharyya feature selection methods for CoDeM, Naïve Bayes, C4.5, and MED classifiers used (Tables 4.4, 4.5, 4.6). In most cases, the Naïve Bayes classifier is found to produce extremely accurate results. This can be explained due to the independent assumption of Naïve Bayes and the low correlated feature sets produced by NASAFS and NASAFS-IDF. For the cases where NASAFS-IDF outperforms the other feature selection methods, the standard deviations indicate a level of significance that can be assigned to these results. To further establish significance, the Wilcoxon signed-rank test is performed. Its results show NASAFS-IDF is a better feature selection method overall, accounting for all classifiers and all data sets, with significance levels of 0.01 to 0.025 (Table 4.7). As previously stated, NASAFS or NASAFS-IDF do not determine a global feature set. However, assumption of a global feature set is possible by using the class feature set that produces the best overall accuracy when used to classify all samples into an appropriate class. When determined in this manner, higher accuracy results are obtained than when using the average of all the class accuracies (Section 4.3).

Overall, NASAFS and NASAFS-IDF accomplish the stated goal of creating a feature selection method to operate in a highly dimensional continuous data domain to provide low-correlated feature sets that produce highly accurate results. NASAFS and NASAFS-IDF also outperform the other feature selection methods evaluated. In some instances, this level of accuracy is extremely high, this is dependent on the data set and classifiers used. Even though NASAFS-IDF is created to work best for hyperspectral data sets, it is shown that it can also produce good classification accuracies when tested on another types of data; these accuracies are found to be as good as, if not better than, the accuracies produced by the common feature selection methods evaluated.

5.2 Recommendations for Future Work

Future work with NASAFS or NASAFS-IDF would require a field data collect and the use of a hyperspectral imager. Collecting this data with known textiles in the FOV will serve a multitude of purposes. It will enable the identification of possible confusers, as well as determine the capability of this system in the field. Using this data, and a predetermined NASAFS or NASAFS-IDF feature set for a specific class, it will be possible to determine that class' degree of detection accuracy, as well as its accuracy against the other textile classes, both known and unknown, in the scene. This would allow for the incorporation of confuser suppression, which will create a more robust system. This will also demonstrate the methods' capability in the presence of pixel mixing (i.e. pixels of less than pure class data).

The heuristic could be adapted to create a true multi-objective optimization of the distributed spacing objective and the discrimination objective. It is possible a multi-objective optimization of these two parameters could produce a feature set that is more robust than the current feature sets created by NASAFS and NASAFS-IDF. A third parameter, variable bin size, could also be included to further enhance optimization.

A possibility for global feature set identification is suggested in this work. However, a more formal methodology should be employed to determine a true global feature set. A true global feature set would distinguish all classes simultaneously, with extremely high accuracy and great robustness, even in the presence of noise. One possible direction for future work would be to combine all feature sets generated for a specific data set, keeping only the features from those sets that are not duplicated. This would obtain all the features that are good discriminators. However, this particular solution has an inherent degree of complexity; it is possible that feature set combination could increase the size of the resulting feature set such that it would no longer be acceptable as a solution set.

Appendix A. Wavelet Decomposition

A wavelet is a periodic oscillation similar to that of a cosine wave; however, it is bounded and does not extend from minus to positive infinity. It starts at zero, increases and decreases in amplitude, then returns to zero (Fig. A.1) [26]. This type of wavelet is useful in engineering and mathematical constructs, because it enables us to extract meaningful information from a signal. For example, if a wavelet is constructed at a specific harmonic frequency, then when this wavelet is convolved with a signal at specific intervals, it will resonate if that frequency is present in the signal. These wavelets are used in wavelet transformation and decomposition to help isolate specific frequencies of interest. Wavelet transformations in the discrete time domain are best understood by visualizing a signal being processed with both a high-pass filter and a low-pass filter (Fig. A.2). This filter pair is determined by the specific wavelet chosen. The high-pass information is considered the *details*; the low-pass information is considered the *approximations*. Typically, the approximation coefficients are considered most relevant; therefore, when dealing with wavelet transformations, only the approximation coefficients are continually being filtered, as shown in Fig. A.3. In this figure, $g[n]$ represents the high-pass filter, and $h[n]$ represents the low-pass filter. After each filter process, a down-sampling occurs; this is denoted in the figure by the circle with the down arrow and the numeral 2. The down-sampling is required to maintain the same total number of dimensions as the original signal. Occasionally, the detail (high-pass) information is equally as important as the approximations (low-pass information); in these cases, wavelet decomposition is performed. The wavelet decomposition is similar to the process for wavelet transformation; however, the detail coefficients for each level are reprocessed with the filtering process, just as the approximations. Fig. A.4 shows this process. The bins of the last level are called leaves; due to the down-sampling, the leaves from this type of decomposition contain half of the coefficients of the original signal filtered.

A two-dimensional wavelet decomposition has been introduced that works well with images [3, 55]. This type of decomposition is computed differently than the

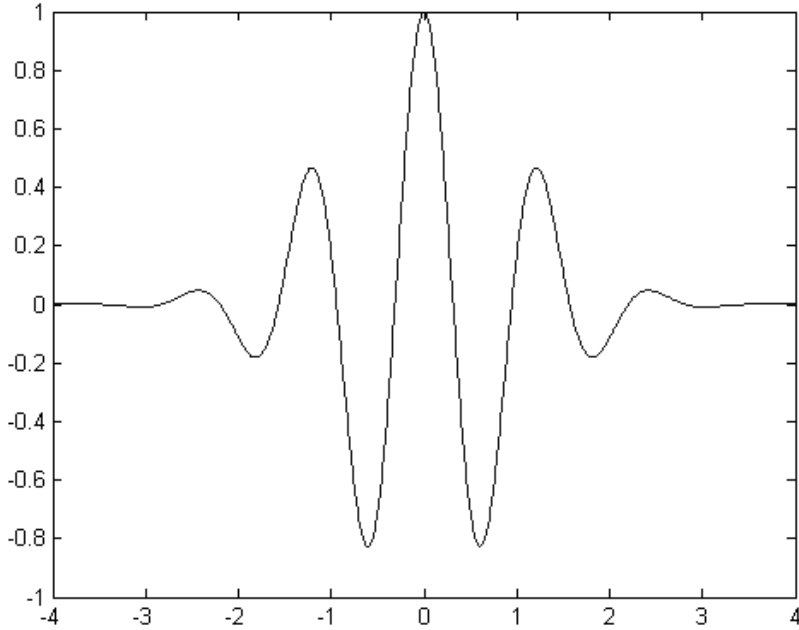


Figure A.1: Example of a Morlet wavelet [91].

wavelet decomposition previously discussed; however, the basic concept still applies. In the two-dimensional wavelet decomposition, three filter techniques are applied to the detail filtering process, instead of one. The detail coefficients are filtered into horizontal, vertical, and diagonal components. The two-dimensional wavelet transform is shown in Fig. A.5. In this figure, cA_n is the approximation coefficient for the n^{th} level, and $cD_n^{(\beta)}$ is the detail coefficient for the n^{th} level, where β is either the horizontal (h), diagonal (d), or vertical (v) component [64]. The two-dimensional decomposition is then created by applying the approximation filters and detail filters to all the coefficients of each level. The texture data set used in this work is processed with the two-dimensional wavelet decomposition using the Daubechies wavelets, specifically the db8, as shown in Fig. A.6 [71]. The leaves are processed via an entropy calculation for this work; however, many different types of statistical methods could be used.

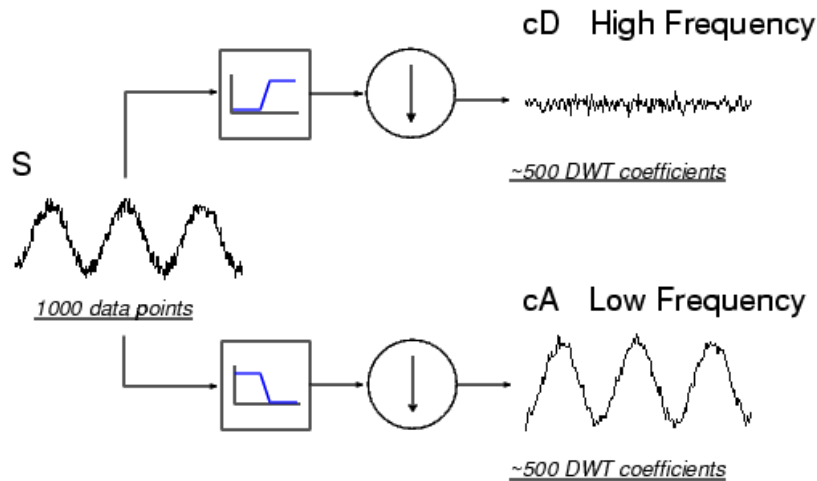


Figure A.2: Example of a signal processed by a generic wavelet transformation. The top portion is the high-pass filter; the bottom portion is the low-pass filter [63].

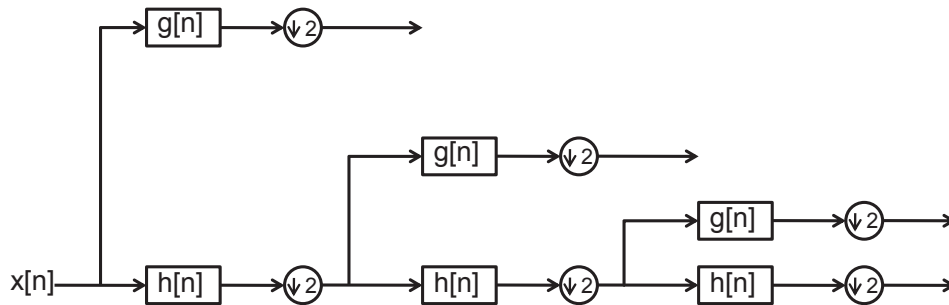


Figure A.3: Example of a wavelet transformation, where $h[n]$ represents the low-pass filter, and $g[n]$ represents the high-pass filter.

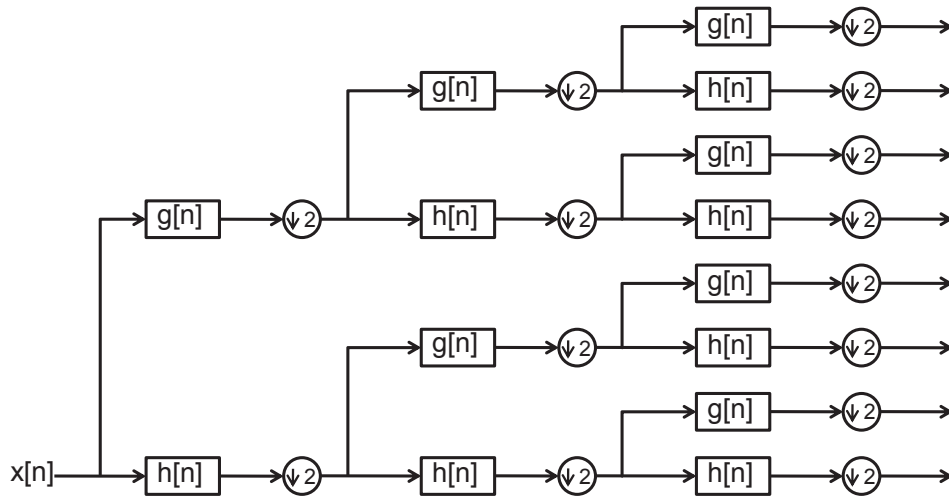


Figure A.4: Example of a wavelet decomposition, where $h[n]$ represents the low-pass filter, and $g[n]$ represents the high-pass filter.

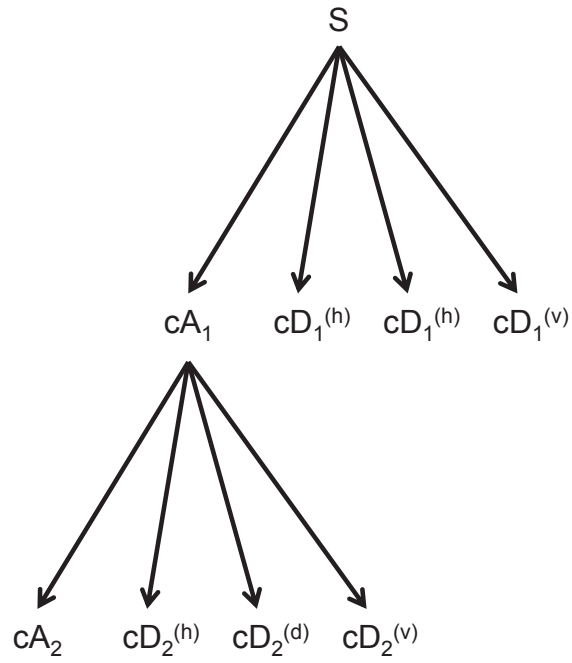


Figure A.5: Example of a two-dimensional wavelet transform, where cA_n is the approximation coefficient for the n^{th} level, and $cD_n^{(\beta)}$ is the detail coefficient for the n^{th} level. β is either the horizontal (h), diagonal (d), or vertical (v) component [64].

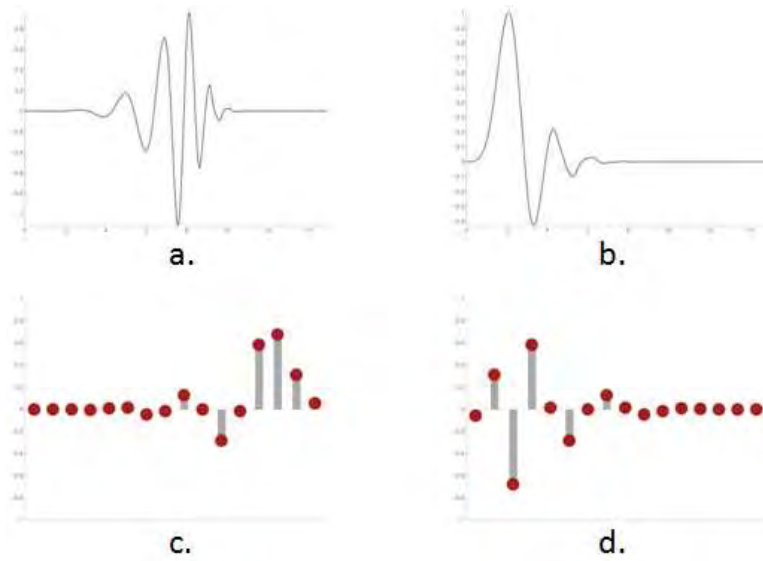


Figure A.6: Daubechies wavelet of the 8^{th} order, where a.) is the wavelet function, b.) is the scaling function, c.) is the digital low-pass filter, and d.) is the digital high-pass filter [71].

Bibliography

1. A. Nunez, USAF, Major. *A Physical Model of Human Skin and its Application for Search and Rescue*. Technical report, Air Force Institute of Technology, Department of Engineering, Wright Patterson Air Force Base, Ohio, 2009.
2. Anonymous, A. Weapons & War, <http://www.neatorama.com/2007/07/24/multicam-now-thats-camouflage>, and <http://yawoot.com/post/1326>, 2007, 2008.
3. Antonini, M., M. Barlaud, P. Mathiew, and I. Daubechies. “Image Coding using Wavelet Transform”. *IEEE Transactions on Image Processing*, 1(2):205–220, 1992.
4. Bertsimas, D. and J. Tsitsiklis. “Simulated Annealing”. *Statistical Sciences*, 8(1):10–15, 1993.
5. Bhattacharyya, A. “On a measure of divergence between two statistical populations defined by their probability distributions”. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.
6. Bishop, C. *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006.
7. Blum, A. and P. Langley. “Selection of relevant features and examples in machine learning”. *Artificial Intelligence*, 97:245–271, 1997.
8. Cantú-Paz, E. “Feature Subset Selection, Class Separability, and Genetic Algorithms”. *Genetic and Evolutionary Computation Conference*, 2004.
9. Chang, C. *Hyperspectral Imaging: techniques for spectral detection and classification*. Kluwer Academic/Plenum, New York, NY, 2003.
10. Chang, C. *Hyperspectral Data Exploitation: Theory and Applications*. John Wiley & Sons. Inc., Hoboken, New Jersey, 2007.
11. Clark, J., M. Mendenhall, and G. Peterson. “Stochastic Feature Selection with Distributed Feature Spacing for Hyperspectral Data”. *IEEE Hyperspectral Image and Signal Processing (WHISPERS)*, 2010.
12. Coello Coello, C., G. Lamont, and D. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer Science + Business Media, New York, NY, 2nd edition, 2007.
13. Coifman, R. and M. Wickerhauser. “Entropy-based algorithms for basis selection”. *IEEE Transactions on Information Theory*, 38:713–719, 1992.
14. Cover, T. “The Best Two Independent Measurements are Not the Two Best”. *IEEE Transaction on Systems, Man and Cybernetics*, 4:116–117, 1974.

15. Cover, T. and J. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.
16. Dash, M. and H. Liu. *Feature Selection for Classification*. Technical report, Department of Information Systems and Computer Science, National University Of Singapore, Singapore 119260, 1997.
17. Dereniak, E. and G. Boreman. *Infrared Detectors and Systems*. John Wiley & Sons, Inc., 1996.
18. Ding, C. and H. Peng. *Minimum Redundancy Feature Selection from Microarray Gene Expression Data*. Technical report, NERSC Division, Lawrence Berkeley National Laboratory, University of California, Berkely, CA, 94720, USA.
19. Ding, C. and H. Peng. “Minimum Redundancy Feature Selection from Microarray Gene Expression Data”. *Procedures of the Second IEEE Computational Systems Bioinformatics Conference*, 523–528, 2003.
20. Driggers, R., P. Cox, and T. Edwards. *Introduction to Infrared and Electro-Optical Systems*. Artech House Inc., Norwood, MA, 1999.
21. editors, Martingale. *A to Z of Knitting: The Ultimate Guide for the Beginner to Advanced Knitter*. Martingale & Co Inc, 2008.
22. Eismann, M. “Hyperspectral Remote Sensing”, 2006. Class Notes.
23. Everitt, B. *The Analysis of Contingency Tables*. Chapman and Hall, 2nd edition, 1992.
24. Fukunaga, K. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, CA, 2nd edition, 1990.
25. González, J., M. Mendenhall, and E. Merényi. “Minimum Surface Bhattacharyya Feature Selection”. *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, 1–4, 2009.
26. Graps, A. “An Introduction to Wavelets”. *IEEE Computational Science and Engineering*, 2(2):50–61, 1995.
27. Guner, B. and J. Johnson. *Comparison of the Shapiro-Wilk and Kurtosis Tests for the Detection of Pulsed Sinusoidal Radio Frequency Interference*. Technical report, Ohio State University, Columbus, Ohio, May 2007.
28. Guyon, I. and A. Elisseeff. “An Introduction to Variable and Feature Selection”. *Journal of Machine Learning Research*, 3(3):1157–1182, 2003.
29. Hall, M. and L. Smith. *Feature Subset Selection: A Correlation Based Filter Approach*. Technical report, University of Waikato, Department of Computer Science, Hamilton, New Zealand.
30. Hall, M. and L. Smith. “Feature Selection for Machine Learning: Comparing a Correlation Based Filter Approach to the Wrapper”. *American Association for Artificial Intelligence*, 1998.

31. Hammer, B. and T. Villmann. “Generalized Relevance Learning Vector Quantization”. *Neural Networks*, 15:1059–1068, 2002.
32. Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
33. Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science + Business Media, New York, NY, 2001.
34. Hecht, E. *Optics*. Addison Wesley, San Francisco, CA, 4th edition, 2002.
35. Holmstrom, L., P. Koistinen, J. Laaksonen, and E. Oja. “Neural and Statistical Classifiers: Taxonomy and Two Case-Studies”. *IEEE Trans. Neural Networks*, 8(1):5–17, 1997.
36. Imaging, Digital and Remote Sensing Laboratory. *Digital Imaging and Remote Sensing Image Generation*. Technical report, Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, Rochester, NY 14923-5604, 2006-2009.
37. Incorporated, ASD. “FieldSpec 3 Hi-Res Portable Spectroradiometer”. <http://www.asdi.com/products/fieldspec-3-hi-res-portable-spectroradiometer>.
38. J. González, USAF, Captain. *Numerical Analysis for Relevant Features in Intrusion Detection*. Technical report, Air Force Institute of Technology, Department of Engineering, Wright Patterson Air Force Base, Ohio, 2009.
39. Jain, A., R. Duin, and J. Mao. “Statistical Pattern Recognition: A Review”. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
40. Jengo, C. and J. LaVeigne. “Sensor performance comparison of HyperSpecTIR instruments 1 and 2”. *Proceedings IEEE Aerospace Conference*, 1805(3), 2004.
41. Jolliffe, I. *Principal Component Analysis*. Springer - Verlag, New York, NY, 1986.
42. Kira, K. and L. Rendell. “The feature selection problem: Traditional methods and a new algorithm”. *Proceedings of Ninth National Conference on Artificial Intelligence*, 129–134, 1992.
43. Kira, K. and L. Rendell. “A Practical Approach to Feature Selection”. *Assorted Conferences and Workshops*, 249–256, 1992.
44. Kirsopp, C., M. Shepperd, and J. Hart. “Search Heuristics, Case-Based Reasoning and Software Project Effort Prediction”. *Proceedings of the Genetic and Evolutionary Computation Conference*, 2002.
45. Koby, C., R. Gilad-Bachrach, A. Navot, and N. Tishby. “Margin analysis of the LVQ algorithm”. *Advances in Neural Information Processing Systems*, 462–469, 2002.

46. Kohonen, T. *Self-Organizing Maps*. Springer - Verlag Berlin, Heidelberg, 3rd edition, 2001.
47. Koller, D. and M. Sahami. "Toward optimal feature selection". *Proceedings of International Conference on Machine Learning*, 284–292, 1996.
48. Kononenko, I. "Estimating Attributes: Analysis and Extensions of Relief". *Proceedings of the European Conference on Machine Learning*, 171–182, 1994.
49. Kumar, S., J. Ghosh, and M. Crawford. "Best-Bases Feature Extraction Algorithms for Classification of Hyperspectral Data". *IEEE Transactions on Geoscience and Remote Sensing*, 39(7):1368–1379, 2001.
50. Lal, T., O. Chapelle, J. Weston, and A. Elisseeff. *Embedded Methods*. Technical report, Max Planck Institute for Biological Cybernetics, Tübingen, Germany.
51. Langley, P. "Selection of relevant features in machine learning". *Proceedings of the AAAI Fall Symposium on Relevance*, 1–5, 1994.
52. Leon-Garcia, A. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, New York, NY, 1994.
53. Lowry, R. *Concepts and Application of Inferential Statistics*. Vassar College, 1998.
54. Luis, T. *An evaluation of filter and wrapper methods for feature selection in categorical clustering*. Technical report, Department of Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Jordi Girona 1-3 08034 Barcelona, Spain.
55. Mallat, S. "A Theory for Multiresolution Signal Decomposition The Wavelet Representation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.
56. Manolakis, D. *Detection Algorithms for Hyperspectral Imaging Applications*. Project report, Massachusetts Institute of Technology, Lexington, Massachusetts, 2002.
57. Martin, C. *Weaving: Methods, Patterns, and Traditions of the Oldest Art*. Walker & Company, New York, NY, 2005.
58. Martin, R. "Detection and Estimation Class notes", 2009.
59. Mason, M. and I. Coleman. *Study of the Surface Emissivity of Textile Fabrics and Materials in the 1 to 15 μ m Range*. Project report, Block Engineering, Incorporated, Cambridge, Massachusetts, 1967.
60. Mendenhall, M. and E. Merényi. "Generalized Relevance Learning Vector Quantization for Classification-Driven Feature Extraction from Hyperspectral Data". *American Society for Photogrammetry and Remote Sensing*, 2006.

61. Mendenhall, M. and E. Merényi. “Relevance-Based Feature Extraction for Hyperspectral Images”. *IEEE Transactions on Neural Networks*, 19(4):658–672, 2008.
62. Merényi, E. “Precision Mining of High-dimensional Patterns with Self-organizing Maps: Interpretation of Hyperspectral Images”. *Quo Vadis Computational Intelligence: New Trends and Approaches in Computational Intelligence Studies in Fuzziness and Soft Computing*, P. Sincak and J. Vascak Eds., 54, 2000. Available: <http://www.ece.rice.edu/ erzsebet/publications.html>.
63. Microsoft, Matlab R2010b. “Help files: wavelet toolbox: getting started: wavelets, a new tool for signal analysis: discrete wavelet transform”. Software, 2010. Accessed 23 May 2011.
64. Microsoft, Matlab R2010b. “Help files: wavelet toolbox: users guide: advanced concepts: fast wavelet transform (FWT) algorithm”. Software, 2010. Accessed 23 May 2011.
65. N. Soliman, USAF, Capt. *Hyperspectral-Augmented Target Tracking*. Thesis, Air Force Institute of Technology, Wright Patterson Air Force Base, Ohio, 2008.
66. Narendra, P. and K. Fukunaga. “A Branch and Bound Algorithm for Feature Selection”. *IEEE Transaction of Computers*, C-26(9):917–922, 1977.
67. Nunez, A., M. Mendenhall, and H. Bertram. “Hyperspectral Modeling of Human Skin Reflectance in the Visible and Near Infrared”. *IEEE Transactions on Biomedical Engineering*, 2008.
68. Peng, H., F. Long, and C. Ding. “Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.
69. Pizzi, N., M. Alexiuk, and W. Pedrycz. *Stochastic Feature Selection for the Discrimination of Biomedical Spectra*. Technical report, Institute for Biodiagnostics National Research Council Canada, and University of Manitoba, Canada, Winnipeg MB, Canada.
70. Punitha, A. and T. Santhanam. “Feature Space Optimization in Breast Cancer Diagnosis Using Linear Vector Quantization”. *Information Technology Journal*, 6(8):1258–1263, 2007.
71. Pywavelets. “Wavelet Daubechies 8 (db8)”. <http://wavelets.pybytes.com/wavelet/db8/>, 2008-2011. Online: accessed 24 July 2011.
72. Qu, G., S. Hariri, and M. Yousif. “A New Dependency and Correlation Analysis for Features”. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1199–1207, 2005.
73. Quinlan, J. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, revised edition, 1993.

74. Russell, S. and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, 2003.
75. Saito, N. and R. Coifman. “Local Discriminant Bases, in Mathematical Imaging: Wavelet Applications in Signal and Image Processing II”. *SPIE Proceedings*, 2303:2–14, 1994.
76. Scharf, L. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley, New York, NY, 1991.
77. Schott, J. *Remote Sensing: The Image Chain Approach*. 2nd edition. Oxford University Press, USA, New York, NY, 2007.
78. Search, San Bernardino Mountain and Rescue. “Katz Search, Devore Rescue”. <http://www.sbmountainsar.org>.
79. Serpico, S. and G. Moser. “Extraction of Spectral Channels From Hyperspectral Images for Classification Purposes”. *IEEE Transaction on Geoscience and Remote Sensing*, 45(2):484–495, 2007.
80. Shapiro, S. and M. Wilk. “An Analysis of Variance Test for Normality (Complete Samples)”. *Biometrika*, 52:591–611, 1965.
81. Sheskin, D. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall, 2nd edition, 2000.
82. Siegel, S. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, 1956.
83. Smith, K. *A Study of Simulated Annealing Techniques for Multi-Objective Optimisation*. Technical report, University of Exeter, 2006.
84. Srinivas, N. and K. Deb. “Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms”. *Evolutionary Computation*, 2(3):221–248.
85. Srinivas, N. and K. Deb. “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms”. *Evolutionary Computation*, 2(3):221–248, 1994.
86. Tsai, D. and C. Chiang. “Rotation-Invariant Patterns Matching using Wavelet Decomposition”. *Pattern Recognition Letters*, 191–201, 2002.
87. unknown. “Brodatz texture images”. <http://www.ux.uis.no/tranden/brodatz.html>.
88. unknown. “Example of a knit”. http://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/Knitting_plaited_stitches_fabric.png/300px-Knitting_plaited_stitches_fabric.png.
89. unknown. “Example of a Weave”. <http://www.heritageshoppe.com/heritage/essays/images/weave.jpg>.

90. Weston, J., S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. “Feature Selection for SVMs”. *Advances in Neural Information Processing Systems*, 2001.
91. Wikipedia. “Wavelet-Morlet.png”. http://en.wikipedia.org/wiki/File:Wavelet_-_Morlet.png, 2005. Online: accessed 23 May 2011.
92. Wilcoxon, F. “Individual Comparisons by Ranking Methods”. *Biometrics Bulletin*, 1(6):80–83, 1945.
93. Yu, L. and H. Liu. “Efficient Feature Selection via Analysis of Relevance and Redundancy”. *Machine Learning Research*, 5:1205–1224, 2004.

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 15 Sept 2011	2. REPORT TYPE Dissertation	3. DATES COVERED (From - To) Sept 2008 - Sept 2011
--	---------------------------------------	--

4. TITLE AND SUBTITLE Distributed Spacing Stochastic Feature Selection and its Application to Textile Classification	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S) Clark, Jeffrey, D., LtCol.	5d. PROJECT NUMBER 09ENG320
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way Wright-Patterson AFB OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/DEE/ENG/11-05
--	---

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/Ryat Olga Mendoza-schrock, Research Mathematician 2241 Avionics Circle Area B Bldg. 620 WPAFB, OH 45433 olga.mendoza-schrock@wpafb.af.mil	10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/Ryat
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A. Approved for Public Release; Distribution Unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Many situations require the need to quickly and accurately locate dismounted individuals in a variety of environments. In conjunction with other dismount detection techniques, being able to detect and classify clothing (textiles) provides a more comprehensive and complete dismount characterization capability. Because textile classification depends on distinguishing between different material types, hyperspectral data, which consists of several hundred spectral channels sampled from a continuous electromagnetic spectrum, is used as a data source. However, a hyperspectral image generates vast amounts of information and can be computationally intractable to analyze. A primary means to reduce the computational complexity is to use feature selection to identify a reduced set of features that effectively represents a specific class. While many feature selection methods exist, applying them to continuous data results in closely clustered feature sets that offer little redundancy and fail in the presence of noise. This dissertation presents a novel feature selection method that limits feature redundancy and improves classification. This method uses a stochastic search algorithm in conjunction with a heuristic that combines measures of distance and dependence to select features.

15. SUBJECT TERMS
^Detection, Dimensionality reduction, Feature selection, Hyperspectral, Machine learning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 149	19a. NAME OF RESPONSIBLE PERSON Michael J. Mendenhall, Maj.
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (937) 255-3636 x4614 michael.mendenhall@afit.edu