IMPROVING NON-LINEAR APPROACHES TO ANOMALY DETECTION, CLASS SEPARATION, AND VISUALIZATION

DISSERTATION

Todd J. Paciencia, Major, USAF

AFIT-ENS-DS-14-D-15

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

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, the Department of Defense, or the United States Government.

AFIT-ENS-DS-14-D-15

IMPROVING NON-LINEAR APPROACHES TO ANOMALY DETECTION, CLASS SEPARATION, AND VISUALIZATION

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 in Operations Research

Todd J. Paciencia, B.A., M.S.

Major, USAF

December 2014

AFIT-ENS-DS-14-D-15

IMPROVING NON-LINEAR APPROACHES TO ANOMALY DETECTION, CLASS SEPARATION, AND VISUALIZATION

Todd J. Paciencia, B.A., M.S.
Major, USAF

Approved:

/signed/	4 Dec 2014
Kenneth W. Bauer Jr., Ph.D. (Chairman)	Date
/signed/	4 Dec 2014
James W. Chrissis, Ph.D. (Member)	Date
/signed/	4 Dec 2014
Mark E. Oxley, Ph.D. (Member)	Date

Accepted:

Adedeji B. Badiru , Ph.D. Dean, Graduate School of Engineering and Management	Date

AFIT-ENS-DS-14-D-15

Abstract

Linear approaches for multivariate data are popular due to their lower complexity, reduced computational time, and easier interpretation. In many cases, linear approaches produce adequate results; however, non-linear methods may generate more robust transformations, features, and decision boundaries. Of course, these non-linear methods present their own unique challenges that often inhibit their use.

In this research, improvements to existing non-linear techniques are investigated for the purposes of providing better, timely class separation and improved anomaly detection on various multivariate datasets, culminating in application to anomaly detection in hyperspectral imagery. Primarily, kernel-based methods are investigated, with some consideration towards other methods. Improvements to existing linear-based algorithms are also explored. Here, it is assumed that any classes in the data have minimal overlap in the originating space or can be made to have minimal overlap in a transformed space, and that class information is unknown *a priori*. Further, improvements are demonstrated for global anomaly detection on a variety of hyperspectral imagery, utilizing fusion of spatial and spectral information, factor analysis, clustering, and screening. Additionally, new approaches for n-dimensional visualization of data and decision boundaries are developed.

Acknowledgments

I would like to thank my family and friends, for their constant support as I engaged in a battle of wills with this endeavor and myself. I would also like to thank my committee members for supporting me with assistance and patience, and enabling me to complete this in my own unique and likely bizarre way.

My biggest thanks extends to Dr. Bauer, for keeping me positive, allowing me to chase many a rabbit down holes as I insisted on doing, and for the often timely insights. I generated many ”hydra” despite knowing not to do so, and it admittedly took me some time to understand that simpler methods can be just as powerful as their complex, extravagant counterparts. I hope that this final result is something that you can be proud of.

A final thanks to Trevor. Both for worldly insights..., and for having patience the many times I asked you a random question about an old dataset or algorithm as if you had generated them.

Todd J. Paciencia

Table of Contents

Page

Page

Page

List of Figures

List of Tables

IMPROVING NON-LINEAR APPROACHES TO ANOMALY DETECTION, CLASS SEPARATION, AND VISUALIZATION

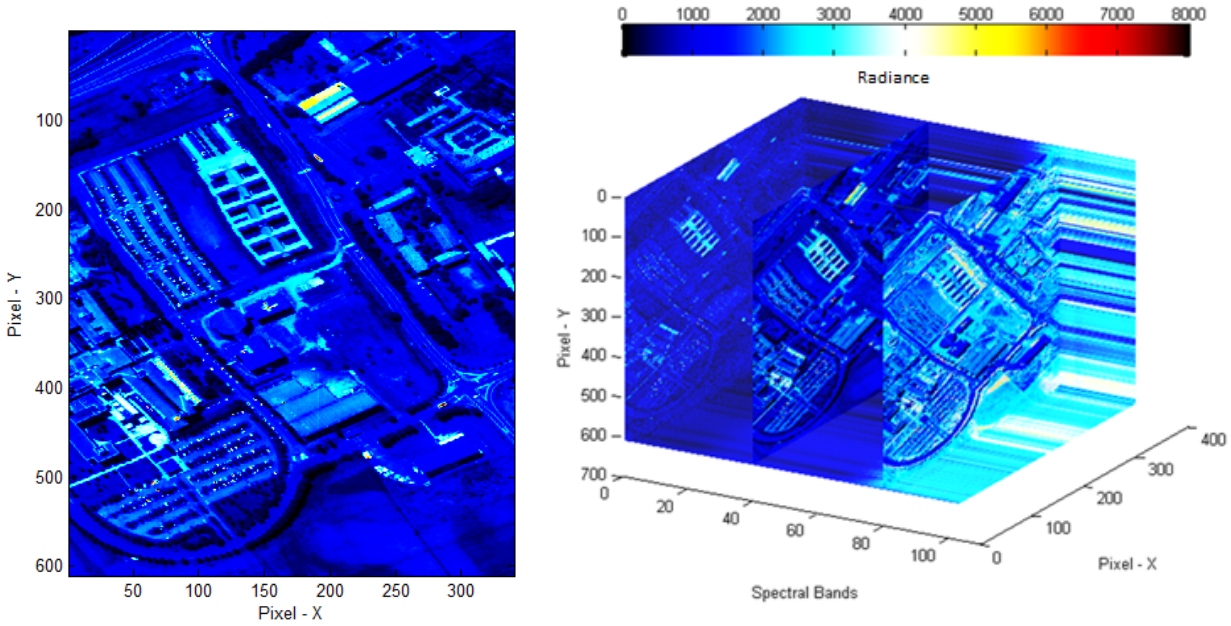
I. Introduction

1.1 Problem Definition and Background

Hyperspectral imagery (HSI) sensors collect contiguous data across the electromagnetic (EM) spectrum, where an area being imaged is divided into a grid with each grid cell or pixel corresponding to a rectangular subregion of the image. HSI sensors record radiance over many discrete intervals, referred to as *spectral bands*, across a subset of optical wavelengths. The sensor records the radiance over the spectral bands for each grid cell or pixel. This generates a cube of band images that contain both spatial and spectral information about the objects and background in a scene. As materials may reflect EM energy differently across the individual wavelengths in comparison to their surroundings (*e.g.*, camouflage as opposed to foliage), this information may serve to identify possible objects or anomalous cells/pixels by analyzing the spectral signatures. Thus, objects of interest can potentially be found by locating pixels that are statistically different than the background.

The full HSI data cube can be very large depending on the number of spectral bands and pixels in the image. Slices from an example data cube are shown in Figure 1.1. This volume of information lends itself to the application of multivariate techniques, but can also have a computational disadvantage without the use of dimension reduction or feature selection. This is also a major consideration for real-time analysis, considering that a goal may be to analyze the data as it is collected by the sensor. As a result, linear transformation

or scoring methods have proven popular for HSI analysis. This is discussed further in Section 3.11.



(a) Single Band.

(b) HSI Cube.

Figure 1.1: HSI Image Radiance Example.

It should be clear that HSI poses several challenges, both due to the amount of data and the nature of the data. The radiance maps corresponding to neighboring wavelengths or bands are often highly correlated. This can cause issues within analytic techniques, or simply is a source of redundant information. Spatial correlation can also exist as the signature of terrain does not change wildly among some neighboring pixels. Images may contain several distinct background classes, resulting in "soft" anomalies when a pixel from one background class is compared to a pixel from another, or when sub-pixels contain different terrain or materials. Further, matrices derived from all pixels, *i.e.*, covariance or

kernel matrices, can be expensive to compute and may need to be approximated in some fashion.

Detecting and identifying an anomaly in HSI can be thought of in two stages: 1) detecting the anomaly, and 2) identifying whether or not the anomaly is actually a target or natural clutter. This research focuses on both stages, although not necessarily independently. That is, it is desirable to have some process or transformation that makes it easy to distinguish a target anomaly from background and natural clutter. New spectral anomaly detection algorithms and enhancement to existing techniques are investigated, where it is assumed there is no prior spectral information about the pixels of interest *a priori*.

1.2 Assumptions

Prior to undertaking this research, some assumptions were made about the images to be processed. First, it must be assumed that the anomalies to be identified are sparse (or at least not dense across the majority of the image), allowing for unsupervised methods to yield a detector. This enables the treatment of detection as a search for rare pixels whose information significantly differs from the local or global background. In the pursuit of enhanced algorithms, other multivariate data sets are used to test and explore various concepts. In these cases, it is assumed that classes within the dataset can be transformed in some manner so as to have reasonable decision boundaries. That is, a level of discrimination needs to be possible.

In reality, the radiation signal reaching the sensor is not as simple as described previously; there are three components: 1) reflected radiance, 2) adjacency radiance, and 3) path radiance. Atmospheric correction calculates and removes the adjacency and path radiance, while retrieving apparent pixel reflectance from the reflected radiance [33, 154]. For purposes of this research, the second assumption is that the data is this derived apparent surface reflectance data, as given by established test images. In these data sets, the spectrum

of each pixel can be viewed as a vector of radiance values for the number of spectral bands. This vector can then be treated as a vector of features, or as a signal representing the pixel due to the large number of bands. Even this is an over-simplification of the HSI data, but other issues are addressed in more detail in Sections 2.3.1 and 4.5.

Third, it is assumed that performance is preferable to speed, although efficiency is a consideration. That is, some level of complexity is allowed in order to achieve gains in identifying anomalies correctly, assuming methods used do not make image analysis computationally intractable.

Finally, it is assumed that the image can be processed in total. The alternative would be to treat the image as though only lines or segments of the image under consideration are available to an analyst at any time due to the receive process from the sensor. In this case, if such a process were to exist, it is assumed that the image sizes used here are representative.

1.3 Research Objectives

The process of finding anomalies has been done many different ways and under many assumptions in the literature. Many of these methods are discussed in the subsequent chapters. Despite the numerous approaches that have been taken, there is a lack of a flexible, yet robust detection algorithm for arbitrary imagery. That is, some of the better-performing algorithms that currently exist show varying performance once different sensors, scene complexity, and/or anomaly density are considered. Further, many state-of-the-art algorithms are limited to linear transformations or decision boundaries. A primary purpose of this research is to explore enhancement or replacement of such algorithms by incorporating non-linear methods. The development of proper means with which to employ some of these non-linear methods is also an important part of this research. As van der Maaten and van den Herik stated [149], although non-linear dimension reduction techniques outperform their linear counterparts on certain complex artificial tasks, successful applications on natural data sets have been scarce. Additionally, simple

adjustments to existing algorithms are explored to find easy performance gains or removal of unnecessary parameters.

1.3.1 Purposeful Visualization of High-Dimensional Data.

The total of this research can be generalized to three areas. First, the HSI data itself is analyzed before any transformation or reduction. To do this, in part, n-dimensional visualization techniques are used and developed. These same techniques are also used as a means to visualize class boundaries and dataset complexity.

1.3.2 Improved Class Separation.

The second area of this research involves using various non-linear methods to generate better unsupervised decision boundaries or spaces for the data under consideration. As mentioned previously, many non-linear methods add complexity and computational expense. Specifically, kernel methods can increase dimensionality, present scaling issues for certain high-dimensional data, and perform very differently depending upon the choice of kernel. In this vein of research the following are investigated:

1. Improved training set generation for Support Vector Data Description (SVDD).
2. Expanding kernel-based methods to the unsupervised case.
3. Selecting an optimal kernel for the kernel methods.
4. Skeleton generation for kernel-based methods.
5. Outlier sensitivity for the resulting non-linear methods.
6. Component selection for Kernel Principal Component Analysis (KPCA).

Here, investigations using other multivariate data sets are first performed to provide crucial information towards translating these methods to larger-dimensional HSI. Additionally, component selection is simultaneously an area of interest for the linear-based methods that use components.

1.3.3 Improvements to Global Anomaly Detection in Hyperspectral Imagery.

Although with HSI data it is known what signatures of different materials look like in the spectrum, in a real-world collection it is not always known what materials may be in the scene. Hence, in anomaly detection it can be more important to seek anomalies without using any kind of spectral matching, at least initially. Numerous algorithms have been developed for this purpose, but they often do not generalize beyond a certain set of scenes, may be difficult to reproduce, or contain several user-defined parameters. Endmember extraction is a related area, where the pixels are treated as combinations of some set of source members. However, even once the endmembers are estimated, a method for anomaly detection must still be performed. Therefore, the focus here is on finding unsupervised factors or transformations that make detection easier, rather than an intense focus on unmixing the pixels. In this research, the lessons learned from the previous areas of research and adjustments to already existing algorithms are all explored. These methods are explored in order to simplify or make existing algorithms more robust. A focus is also to reduce the number of parameters necessary to adapt algorithms to varying image types. The use of a fusion of spatial, spectral, and signal-to-noise information, as well as factor analysis is investigated to provide a better global anomaly detection framework.

1.4 Dissertation Outline

The research that follows is very inter-related, but an attempt is made to present it as linearly and sensibly as possible. First, the data sets used are presented in Chapter 2. Next, some general methods that apply across areas or that recur, or that are traditionally used heavily for a topic, are presented in Chapter 3. Investigation into the HSI data sets themselves and identification of noisy features and bands is shown in Chapter 4. Chapter 5 includes a literature review of n-dimensional methods and development of adjustments to provide more useful visualizations. Chapter 6 incorporates findings from Chapter 4, as well as other methodologies, towards the development of an improvement to existing

anomaly detection techniques. Chapter 7 discusses Kernel Principal Component Analysis (KPCA), and investigates its use as an efficient replacement for linear methods. Chapter 8 includes a review of SVDD literature and the development of an algorithm for better, pseudo-optimal two-class separation or anomaly detection. Finally, Chapter 9 provides a summary of contributions and possible areas for future research.

II. Overview of Data Sets

This Chapter discusses the different data sets that are analyzed throughout this research. These data sets include both natural data and HSI, are real-valued, and are of varying dimensions. Some have only two or three features so that they can be plotted for interpretation, while most are n-dimensional, in that they have greater than three features. Here, features correspond to the variables in the data and exemplars correspond to the observations, or in terms of vector notation, each exemplar is a vector and each feature is a component. For HSI, the bands may be interpreted as the features and each pixel is an exemplar. Classes of data typically exist within each dataset, such as background and anomaly classes in HSI.

2.1 Data Imputation

A few of the data sets used for this research have missing values or values that cannot occur naturally. When exemplars are missing correct feature information, data imputation can be used to replace these values using information already found in the full dataset. For this research, a form of nearest-neighbor imputation was used for the applicable data sets that were not hyperspectral. Let X denote the $N \times p$ data matrix, where there are N exemplars and p features. Specifically, the following steps were performed:

1. Check for *Not a Number* (NaN) values, or values that do not occur naturally, in the dataset. Return if none found.

2. Let $\boldsymbol{x}_i$ denote the i-th exemplar in dataset X. Normalize the dataset by feature using $x_{ij}^{norm} = \dfrac{x_{ij} - \mu_{*j}}{\sigma_{*j}}$, where $*$ indicates over all exemplars in X in feature j, and μ and σ denote the mean and standard deviation, respectively.

3. For an exemplar $\boldsymbol{x}_i$ with missing data and of class type c, let X_c be the set of exemplars with no missing data and also of class type c, not including $\boldsymbol{x}_i$. Let X_c^{norm} be the corresponding normalized data.

4. Define the nearest neighbor $\boldsymbol{y}$ to $\boldsymbol{x}_i$ as that corresponding to:
$argmin_{\{\boldsymbol{y}^{norm} \in X_c^{norm}\}} \sum_j |x_{ij}^{norm} - y_j^{norm}|$, where j is only used if that feature is not missing.

5. Replace the missing feature values in $\boldsymbol{x}_i$ with the values from $\boldsymbol{y}$.

In other words, it is assumed that data with missing feature information is sparse and that any exemplar in the set being imputed has a similar existing neighbor. The L_1 norm, in conjunction with the normalization, is used to ensure that a neighbor is not identified where a large deviation occurs in some feature, while taking into account all deviations. This simple imputation provides a quick technique to replace missing values. Alternate forms of imputation do exist, other than using a different distance metric [183]. For example, imputation by regression prediction assumes continuous features, which is an erroneous assumption for the data sets that were imputed in this research. An additional method is to replace missing data with the mean of the feature for the corresponding class. This latter method, however, could skew the exemplar closer to the centroid of the dataset and could also provide a feature value not physically or naturally possible.

2.2 Multivariate

This section describes the multivariate data sets used in this research to test various algorithms. These were chosen due to known properties, or issues that they can present to algorithms.

2.2.1 Breast Cancer Wisconsin (Diagnostic).

The *Breast Cancer Wisconsin (Diagnostic)* dataset contains 699 exemplars with nine tumor features [19]. These nine features are: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland

chromatin, normal nucleoli, and mitoses. Tumors are classified as malignant or benign, and the dataset in its original form has sixteen missing values. To correct these missing values for this research, nearest-neighbor imputation, within class, is used as was discussed in Section 2.1.

2.2.2 Chainlink.

The *Chainlink* dataset used is that taken from Burk [41], and has 500 exemplars in each of two classes that form linked rings. This dataset is good for the investigation of algorithm performance given a certain data geometry. The data is depicted in Figure 2.1.

2.2.3 Modified Banana.

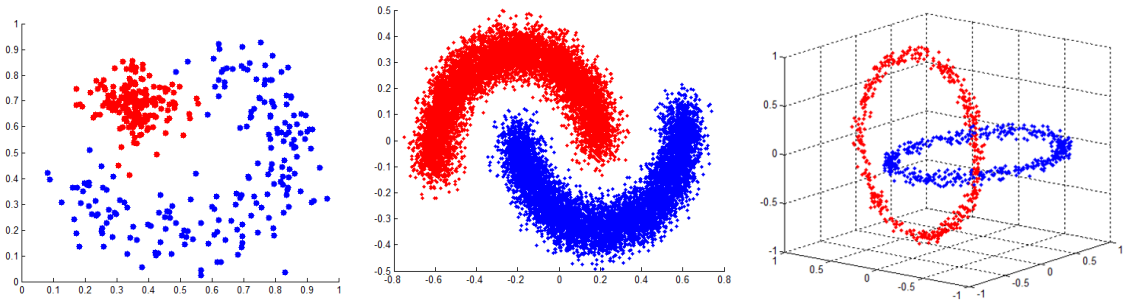
The *Modified Banana* dataset is not a standard dataset, and was specially constructed for this research. A banana dataset typically consists of a class of data taking a crescent shape, and another challenging the class boundary in some manner. Here, one class of data was constructed so as to have a fairly crescent shape, while also having an imperfect boundary. Further, a second class was added so as to be within the crescent. Here, the intent is to give a non-linear classifier difficulties should any data from the second class erroneously be used to find the boundary, or should a too-perfect crescent shape be used. The dataset consists of 400 exemplars, with 200 in each class. This set is shown in Figure 2.1.

2.2.4 Half-Moons.

The *Half-Moons* data, specifically that taken from Burk [41], has approximately 7500 exemplars in each of two classes. The classes form two non-overlapping crescents. This dataset is also good for investigating algorithm performance given a certain data geometry. It is also very similar to the banana dataset.

2.2.5 Fisher Iris.

Fisher Iris is a popular dataset in pattern recognition literature due to one class being separable from the other two and the latter not being linearly separable. The dataset



(a) Banana. (b) Half-Moons. (c) Chainlink.

Figure 2.1: 2-Class Geometric Problems.

contains three classes of fifty exemplars each. Sepal length, sepal width, petal length, and petal width, all in cm, are features for iris setosa, versicolour, and virginica flower types [19].

2.2.6 Pima.

The *Pima Indian Diabetes* dataset contains 768 exemplars with twelve features [19]. All patients are females at least 21-years old of Pima Indian heritage. The twelve features are: number of times pregnant, plasma glucose concentration at two hours in an oral glucose tolerance test, diastolic blood pressure (mm Hg), triceps skin fold thickness (mm), two-hour serum insulin (mu U/ml), body mass index (weight in kg/(height in m)^2), diabetes pedigree function, and age (years). Patients are classified as having tested positive or not for diabetes.

Although the dataset contains no missing values, there are zeros in places where they are biologically impossible [19]. These zeros occur erroneously in the plasma glucose concentration, diastolic blood pressure, and body mass index variables [93]. To correct these erroneous values for this research, nearest-neighbor imputation, within class, is used as was discussed in Section 2.1. 44 exemplars required imputation, but only seven of these required imputation for more than one feature.

2.2.7 Vertebral Column.

The *Vertebral Column* dataset contains 310 exemplars with six features [19]. The six biomechanical features derived from the shape and orientation of the pelvis and lumbar spine are: pelvic incidence, pelvic tilt, lumbar lordosis angle, sacral slope, pelvic radius, and grade of spondylolisthesis. The dataset can be split into two or three classes. The two-class version classifies by normal and abnormal, while the three-class version splits the abnormal class into disk hernia and spondylolisthesis classes.

2.2.8 Hepta.

The *Hepta* dataset is a seven-class dataset for geometry investigation or clustering testing. Each class consists of approximately 30 data points, where six of the classes surround one. This dataset is shown in Figure 2.2 and was also taken from Burk [41].

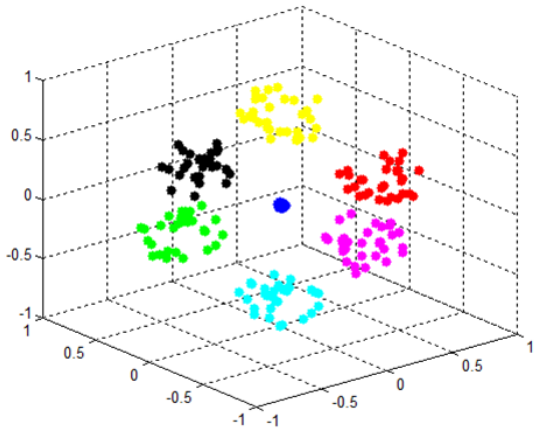


Figure 2.2: Hepta dataset.

2.2.9 Wine.

The *Wine* dataset is the result of chemical analysis of wines grown in a region in Italy derived from three different cultivars. Thirteen attributes were measured to classify

these three cultivars: alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and proline. This dataset is considered well-behaved for class structures [19].

2.2.10 Wine Quality.

The *Wine Quality* dataset consists of 6,497, eleven-feature exemplars, where each exemplar corresponds to a red or white variant of the Portuguese "Vinho Verde" wine [60]. The eleven features used (based on physicochemical tests) are: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, suplhates, and alcohol. In addition to classifying as red or white, a quality (score between 0 and 10) output variable is also included. Unique values for this score lie between 3 and 9.

2.2.11 MNIST.

The *MNIST* database of handwritten digits has a training set of 60,000 exemplars and a test set of 10,000 exemplars [3, 4]. The digits are size-normalized and centered in a fixed-size 28 pixel-squared image, where each image has been vectorized to have 784 pixels/features. An example digit image is shown in Figure 2.3. 65 pixels have no value (black) over all exemplars and both the training and test sets. As these provide no information, they were removed from the dataset, yielding a final 719 features.

2.2.12 Arcene.

The *Arcene* dataset was a part of the Neural Information Processing Systems (NIPS) 2003 feature selection challenge, and is a two-class problem with continuous input variables. Specifically, the data is mass-spectrometric obtained using surface-enhanced laser desorption ionization and was combined from National Cancer Institute and Eastern Virginia Medical School sources [19]. The data was pre-processed before release so as to build a valid benchmark, to include putting variables on a common range. The positive class

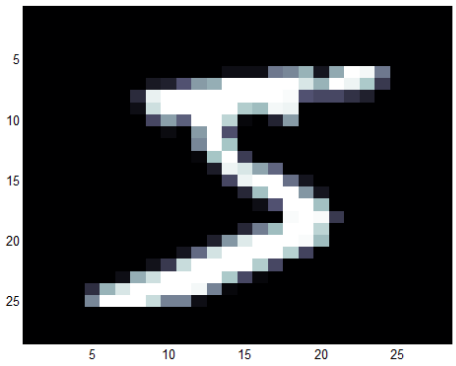


Figure 2.3: Example MNIST Digit: 5.

includes patients with ovarian or prostate cancer, while the negative includes healthy or control patients. 7,000 of the 10,000 features are real, indicating an abundance of proteins in human sear having a given mass value. The remaining 3,000, referred to as probes, were added as distractor features and have no predictive power. The order of all 10,000 features was randomized.

The dataset includes 100-exemplar training and validation sets with full class information, and a 700-exemplar test set where originally it was only known that 310 of the 700 exemplars were positive. For this research, the training and validations sets were combined to generate a dataset, where 88 of the 200 exemplars are positive. Additionally, a 900 exemplar dataset was generated by adding the 700 exemplar test set and its corresponding class labels [91]. The Arcene data was chosen for this research for several reasons: its distractor features, the fact that the exemplars can be treated as signals, its difficult discrimination, and that the number of features is significantly larger than the number of exemplars. The mean vectors for both exemplar classes are shown in Figure 2.4. Because the order of features was randomized, these no longer represent exemplar signatures, as they would have if properly ordered.

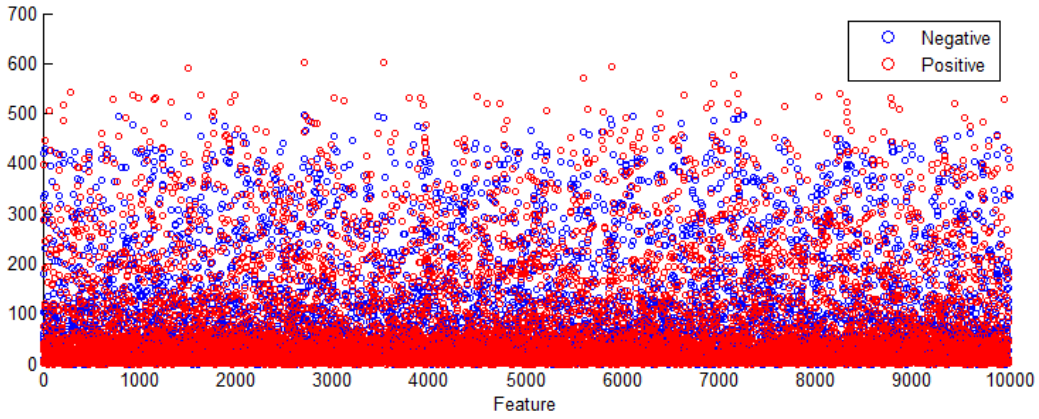


Figure 2.4: Arcene Class Mean Vectors.

2.3 Hyperspectral Imagery

For a HSI image, each channel or band wavelength of a pixel is typically in micrometers or microns (μm), where this measurement represents the radiance. The peaks and valleys of a pixel's spectrum, not due to the Sun or the atmosphere, reveal information on the chemical composition of the pixel under examination. Images from a variety of sensors are used in this research.

2.3.1 Special Considerations for HSI.

HSI can present unique issues in comparison to other multivariate data sets. These include atmospheric properties, scaling issues, and truth mask issues. Further, sensor collection or correction error can cause occasional bad values in the data. One specific example of this occurs in the HYDICE ARES imagery (Section 2.3.2), where negative values occur in a sparse manner. As this is infrequent in the data, and because past researchers have done the same [107, 111], here such data is set to zero. Correlation can also present an obstacle for certain algorithms.

2.3.1.1 Atmospheric Properties.

Spectral reflectance is the ratio of reflected to incident energy as a function of wavelength. This varies with wavelength for many materials because energy at certain wavelengths is scattered or absorbed to different degrees [193]. The locations of spectral regions used to retrieve atmospheric and physical features are shown in Figure 2.5 for reference. All natural materials exhibit some variability in composition and structure that similarly yields variability in the reflectance spectra. Low values on spectral signatures indicate wavelength ranges for which materials absorb the incident energy; these bands are commonly called *absorption bands*.

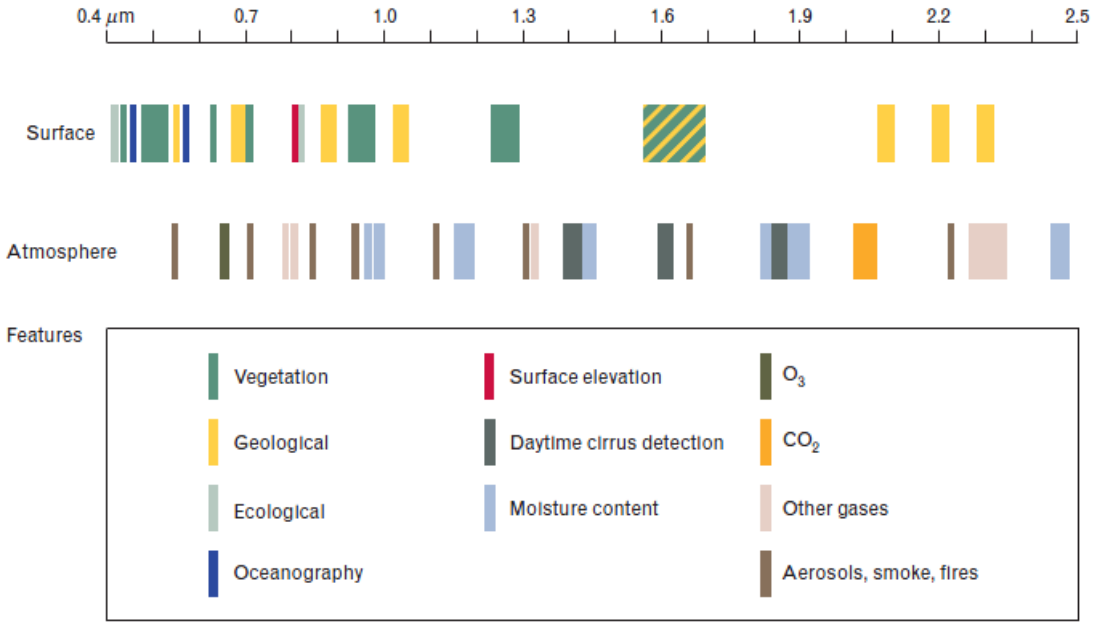


Figure 2.5: Spectral Region Locations [81].

Measured reflectance by the sensor is affected by more than just the spectral reflectance of surface materials and the spectrum of the input solar energy. Interactions of the energy during its downward and upwards passages through the atmosphere,

the geometry of illumination, and characteristics of the sensor system all affect the measurement as well [193]. Of particular interest are the effects of the intervening atmosphere on the surface reflectance estimate. Atmospheric absorption bands are frequency bands at which the energy emitted is almost completely absorbed by the atmosphere. As Smetek stated in his research [191], a sensor detects primarily random noise in such bands.

At wavelengths below 2.5 μm, the incident solar flux is impacted by the absorption by well-mixed gases (such as ozone, oxygen, methane, and carbon dioxide), absorption by water vapor, scattering by molecules, and scattering and absorption by aerosols and hydrometeors. Molecular scattering effects can be significant out to 0.75 μm, and aerosol scattering continues to have an impact at 1.3 μm [81]. Mixed gases can be modeled accurately and data is often corrected accordingly before being given for analysis, as had been done with the images used here. Water vapor absorption has perhaps the most notable remaining effect after such processing has been applied to the images. Two very weak absorption bands are located at 0.6 and 0.66 μm, slightly stronger absorption bands are located at 0.73, 0.82, and 0.91 μm, and at 0.94 and 1.14 μm water absorption is even stronger. It is strongest near 1.375, 1.9, and 2.5 μm, such that retrieval of surface reflectance is very difficult or impossible [81]. Due to this, data collected by the sensor near such bands can be noise. Therefore, it is useful to remove these strong absorption bands from the data cube before conducting analysis. Unfortunately, finding absorption or noisy bands is not always as simple as removing only those bands containing these wavelengths.

Past researchers have in some way interpreted the relative noisiness of each band to decide which bands to remove from an image, or have removed the same as their predecessors. An attempt at developing a more rigorous approach to find absorption and noisy bands is made for the different sensor images in Section 4.5, and identification results are compared against those bands identified in the literature. In general, the strong water

absorption bands mentioned correspond to much of what is removed for the HSI data sets before analysis. Development of the approach was further necessitated by the fact that some of the HSI images used in this research are not prevalent in the literature.

2.3.1.2 Scaling.

Due to the number of bands and the magnitude of the spectral values, in some cases HSI can present unique issues for algorithms. In particular, the dot products between exemplars often found in kernel methods can grow to be too large in scale. To prevent such scaling issues, the hyperspectral data was also scaled by dividing by the maximum value found across all pixels and all spectra. This maintained the shape and relative magnitude of the signatures, while alleviating computational issues when necessary. Such a technique was also used by Kwon and Nasrabadi [133] when using Kernel Principal Component Analysis (KPCA) for a Kernel Reed-Xiaoli (KRX) algorithm. Scaling by a constant has little effect on eigenvalues and eigenvectors for a covariance matrix when using methods such as PCA (discussed in Section 3.2). Rather, it simply scales the eigenvalues, as for a random variable X and constant c, $Var(cX) = c^2 Var(X)$. This effect on the eigenvalues is shown in Figure 2.6. This can cause its own problems, however. The scaled values can become too small for an intended purpose, if trying to use a fixed magnitude-based cut-off for dimension reduction, or the scaled data can become small enough that it causes precision error in the estimation of the eigenvalues and eigenvectors. In any results shown, it is made clear during discussion which version of an HSI dataset is being used, along with any adjustments that had to be made. Alternatively, the data could be standardized. However, that would change the relative information amongst features and thus the relative spectral signatures.

2.3.1.3 Correlation.

Both spatial and spectral correlation can also present issues for algorithms when working with HSI. Spatial correlation occurs because neighboring pixels may contain

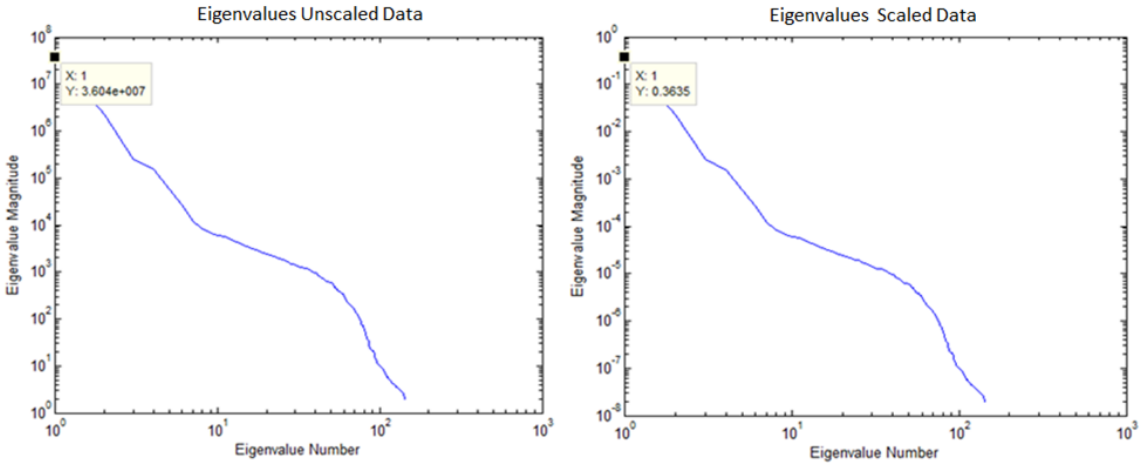
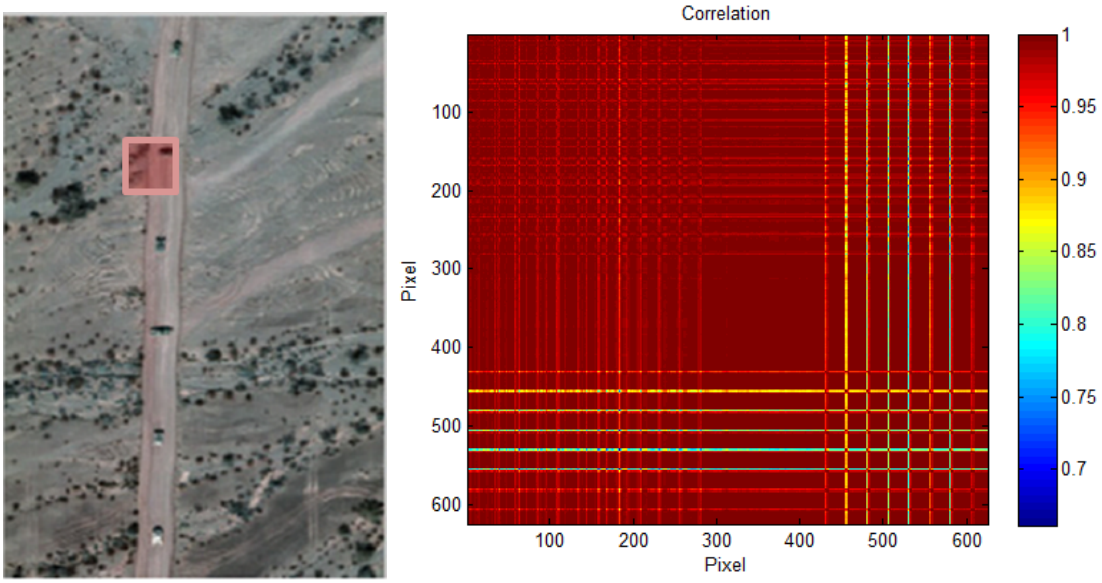


Figure 2.6: ARES1D Covariance Eigenvalues Comparison.

similar materials, and because shapes (versus magnitudes) of pixel spectral signatures are often similar. Consider the 25×25 pixel window and its corresponding correlation matrix shown in Figure 2.7. The window contains brush, road, and vehicle (anomaly) pixels that are all highly correlated.

The similar shape of signatures is shown using another sensor and image in Figure 2.8, where 1,000 pixels were randomly sampled. Although different materials have different magnitudes across the bands, the shapes of these signatures are often similar. The spectral correlation for segments of neighboring bands can also be seen using this same Figure. Those bands where the signatures dip to near zero in many cases correspond to absorption bands. Alternatively, when considering the band correlation matrix, shown in Figure 2.9, and referring back to HYDICE data, it can be seen that bands are highly correlated with their neighbors with some exceptions when an absorption band occurs or a certain new range of the spectrum is reached. The segments just mentioned in the pixel signatures become obvious from this matrix. Miller [160] used this property to determine a reduced set of bands for his anomaly detection algorithm.



(a) Window. (b) Pixel Correlations.

Figure 2.7: ARES1D Window Correlation.

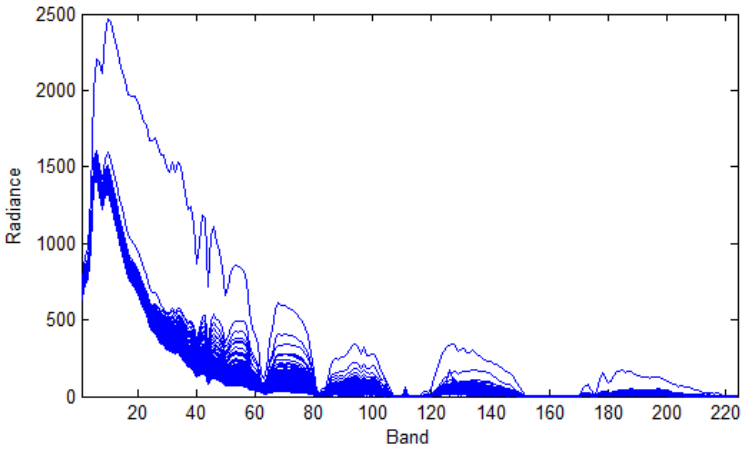


Figure 2.8: AVIRIS Deepwater Scene1 Sample.

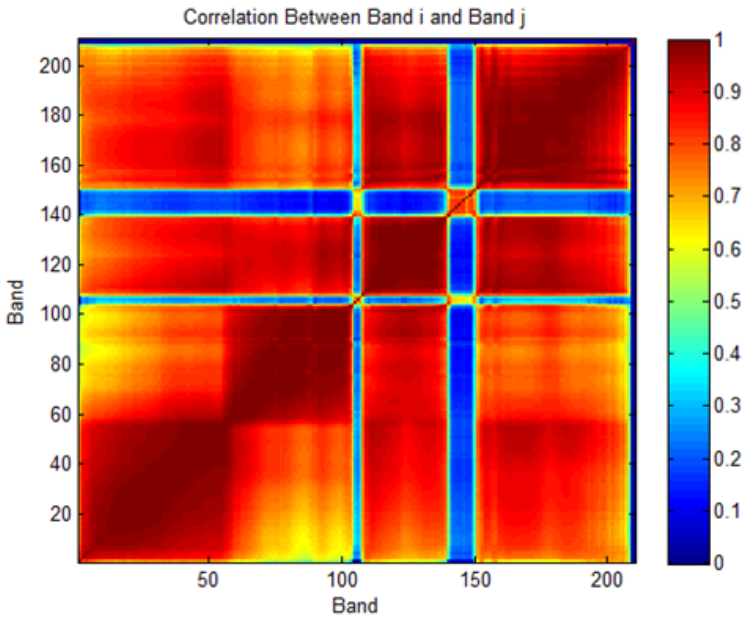


Figure 2.9: ARES1D Band Correlation.

Window-based detection methods can suffer as a result of spatial correlation, as background estimates may be based on pixels that look very much like anomalies. On the other hand, spectral correlation can serve to mask subtle differences between anomalies and non-anomalies.

2.3.1.4 Truth Masks.

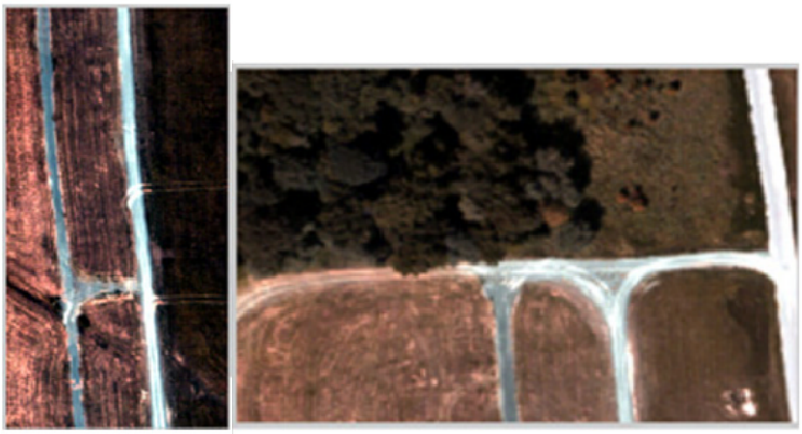
As the real-world size of a pixel increases, so too does the likelihood that more than one material is contributing to that pixel's signature. That is, at higher altitude the sensor is more likely to include several materials within one captured pixel. Therefore, an observed pixel signature may, in fact, be a combination of a number of endmember spectra, or several sub-pixel materials. This possibility can present difficulties in the generation of a truth mask for such images. As a result, the archival truth masks for the ARES HYDICE and HyMAP images reflect such sub-pixel or border anomalous pixels, in addition to the true anomalous pixels. This has allowed researchers to treat these pixels differently when generating true

positives or false positives, making comparisons to previous results or algorithms difficult. In order to standardize how these identified pixels are treated, they are investigated in detail in Section 4.3.

2.3.2 HYDICE-Derived.

It should be noted that some of the data sets used have been passed on by previous researchers. Therefore, in some cases these may be sub-images of an original image and the origin truth masks presented may be a previous researcher's interpretation. Graphics or metrics are included as often as possible to provide clarification.

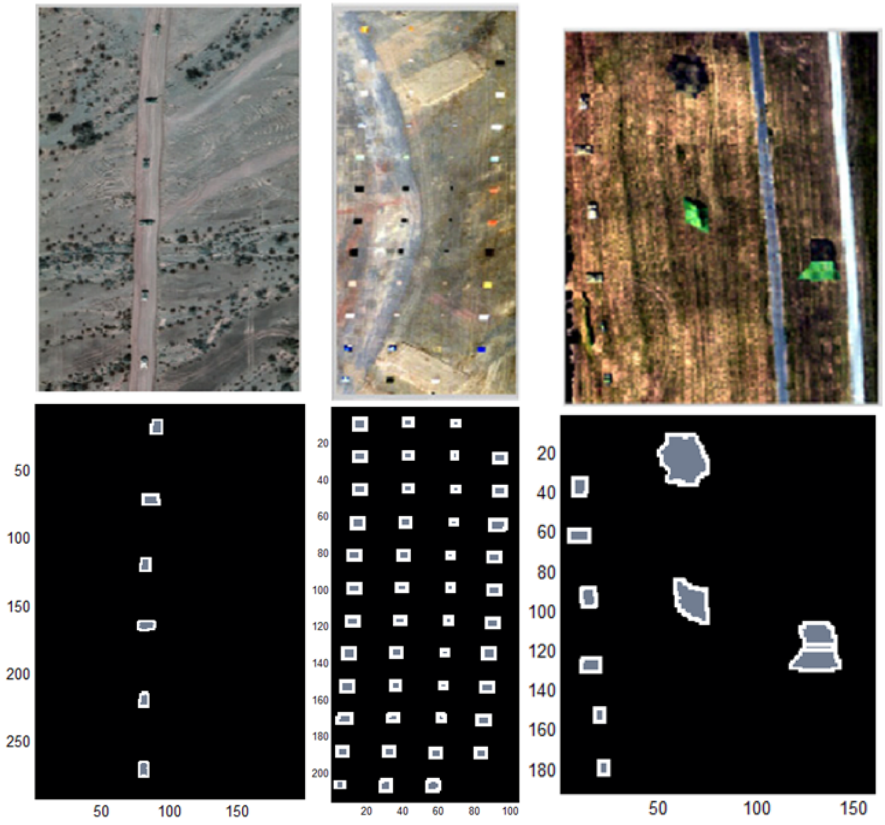
The Hyperspectral Digital Imagery Collection Equipment sensor-derived (HYDICE) images have 210 spectral bands, between 0.397 and 2.5 μm, including visible through short-wave infrared data. Images from this sensor used here are forest or desert-dominated scenes. The HYDICE ARES1C and ARES2C images are from a rural environment with no specific man-made objects of interest. Their corresponding natural images are shown in Figure 2.10.



(a) ARES1C. (b) ARES2C.

Figure 2.10: Natural No-Target HYDICE Images.

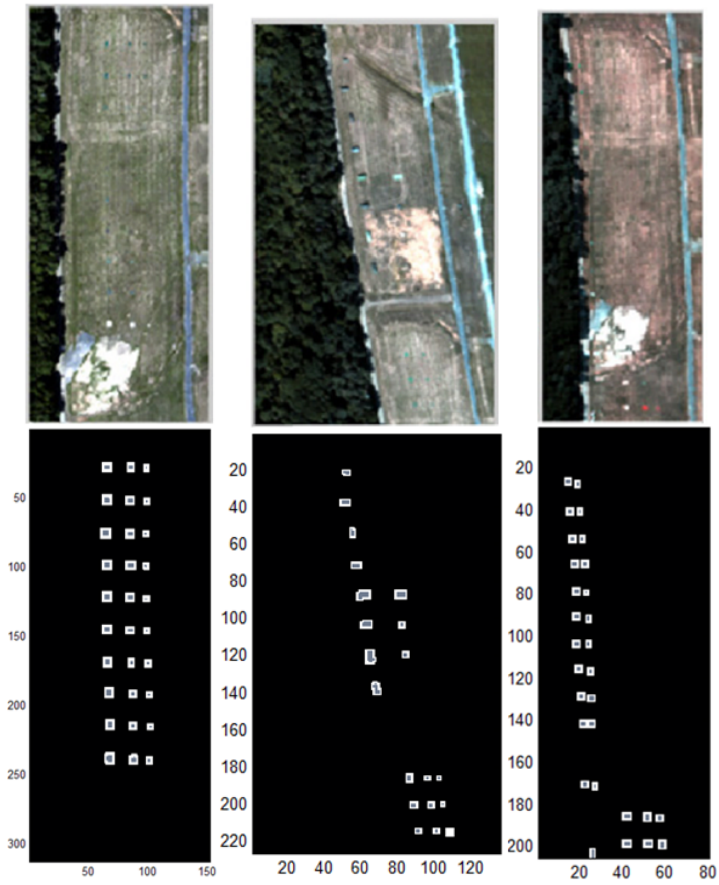
Nearly all of the remaining HYDICE-derived images used are from the Desert radiance II (D) and Forest radiance I (F) collection events. The corresponding set of natural images is shown in Figures 2.11 and 2.12. Most of these images also had target border/neighborhood pixels in their truth masks, depicted here in white. Again, this issue and the proposed resolution is further discussed in Section 3.13.



(a) ARES1D: 6. (b) ARES2D: 46. (c) ARES1F: 10.

Figure 2.11: Three HYDICE Images and Number of Targets.

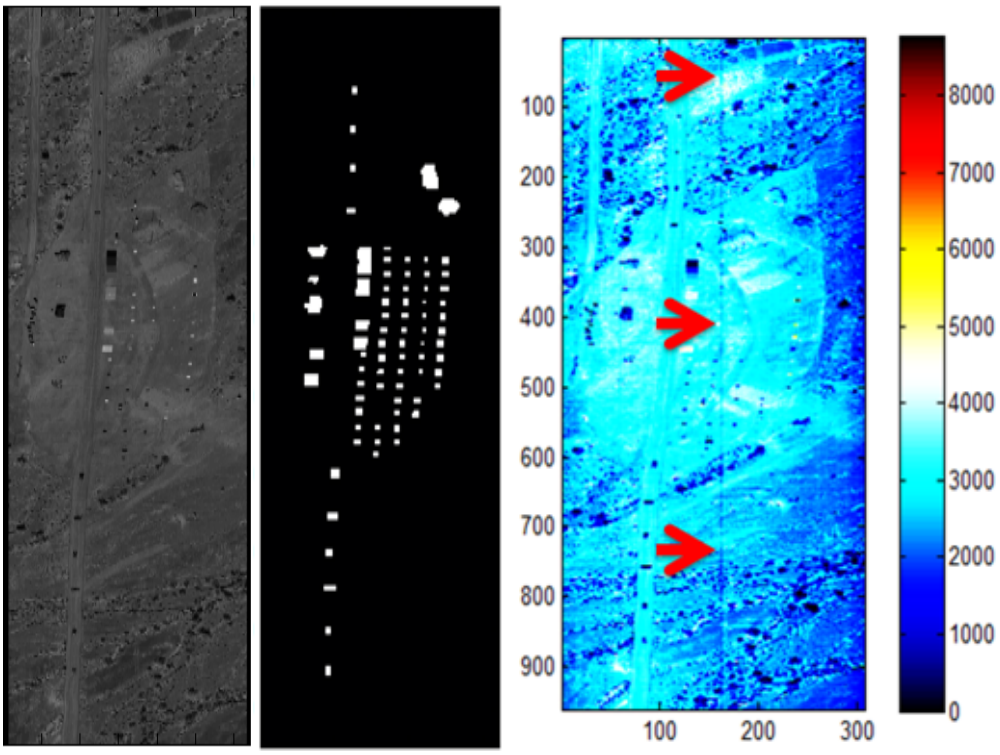
The final HYDICE image used is *run03m20*. This image has no target border/neighborhood pixels in the truth mask, and was taken at 5160.1 feet above ground level (AGL). The first



(a) ARES2F: 30. (b) ARES3F: 20. (c) ARES4F: 29.

Figure 2.12: Three HYDICE Images and Number of Targets.

eight and final five pixel columns from the original image are removed, as they erroneously contain only zero values. The natural image and its truth mask are shown in Figure 2.13. Also shown is one of a few bands in the data where an artifact zero-line occurs. These were retained going into the analysis found in Section 4.5, despite this issue, with the understanding that it would only make classification more difficult. This image is used due to the amount of targets and their close proximity, yielding potential issues for window-based methods. A summary of the HYDICE images is shown in Table 2.1, where anomalous pixels do not include border pixels.



(a) Natural. (b) Truth: 79 Targets. (c) Artifact Line.

Figure 2.13: HYDICE run03m20.

2.3.3 AVIRIS.

The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) data sets are used courtesy of the National Aeronautics and Space Administration (NASA) and the Jet Propulsion Laboratory of the California Institute of Technology. AVIRIS images contain 224 spectral channels between 0.4 and 2.5 μm.

Three Deepwater Horizon images are used, each with associated truth masks developed to correspond with man-made objects in the scene. Scene1 contains 23 targets and is from run *f100517t01p00r11rdn_b_sc01_ort_img*. Ship1 contains 6 targets and is from run *f100710t01p00r08rdn_b_sc01_ort_img*. 4Ships2 contains 4 targets and is from

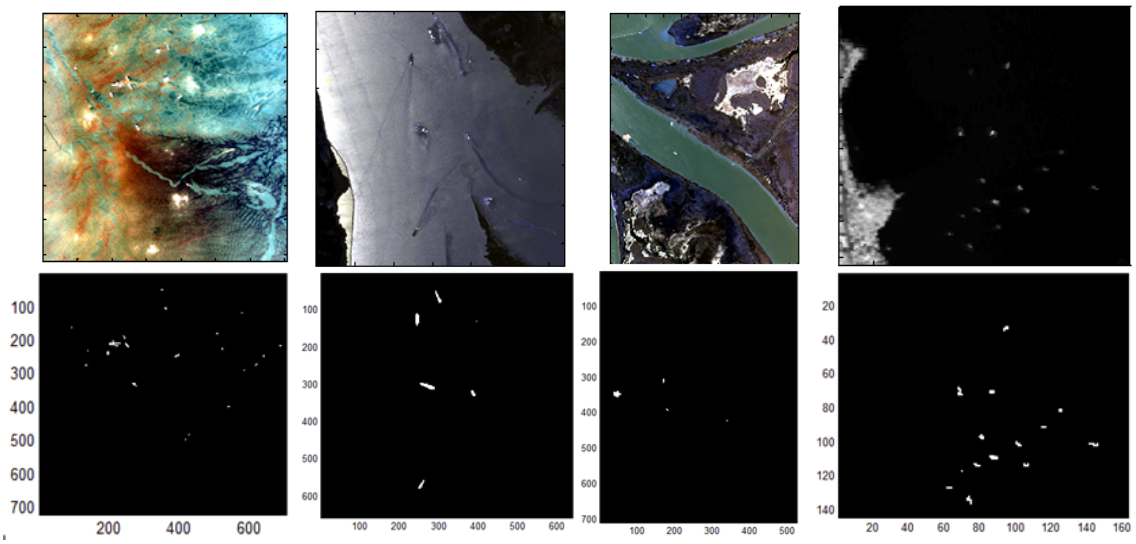
Table 2.1: HYDICE Image Properties.

Image	Dimensions	Pixels	Anomalies	Anomalous Pixels	Border Pixels
ARES1C	203×108	$21,924$	0	0	0
ARES2C	124×198	$24,552$	0	0	0
ARES1D	291×199	$57,909$	6	235	437
ARES2D	215×104	$22,360$	46	523	1942
ARES1F	191×160	$30,560$	10	$1,007$	973
ARES2F	312×152	$47,424$	30	307	1221
ARES3F	226×136	$30,736$	20	145	314
ARES4F	205×80	$16,400$	29	109	339
run03m20	960×299	$287,040$	79	$8,255$	0

run *f100929t01p00r13rdn_b_sc01_ort_img*. A fourth AVIRIS image of the Virgin Islands is used, depicting 14 targets from run *f051219t01p00r14c_sc01_geo_img*. These four images were chosen for the purposes of variety and due to their varying sizes. They also provide a contrast in scene type relative to the HYDICE imagery. The natural images and their truth masks are shown in Figure 2.14. A summary for the AVIRIS images is shown in Table 2.2.

2.3.4 Pavia.

The Pavia data sets are two scenes acquired by the ROSIS sensor during flights over Pavia in northern Italy and were provided by the Telecommunications and Remote Sensing Laboratory of Pavia University [1]. These provide an investigation of urban scenes from a non-HYDICE sensor with ground truth. The Pavia Centre scene is 1096×715 pixels and contains 102 bands, while the Pavia University scene is 610×340 pixels and contains



(a) Scene1. (b) Ship1. (c) 4Ships2. (d) VirginIslands1.

Figure 2.14: AVIRIS Images.

Table 2.2: AVIRIS Image Properties.

Image	Dimensions	Pixels	Anomalies	Anomalous Pixels
Scene1	720×707	$509,040$	23	887
Ship1	657×640	$420,480$	6	$1,025$
4Ships2	709×526	$372,934$	4	332
VirginIslands1	144×163	$23,472$	14	80

103 bands. The bands in both images are between approximately 0.43 and 0.86 μm [5]. In the Pavia Centre image, two collection areas are joined at line 223. The geometric resolution is 1.3 m, and each image contains a set of nine determined classes and an additional background class to serve as ground truth.

In the case of the Pavia Centre scene, the background, water, and meadows classes dominate the scene as approximately 95% of the scene. There is no true outlier class, with remaining classes having between 2600 and 9200 pixels each. The Pavia Centre scene, ground truth, and a small subset of randomly sampled class signatures are shown in Figure 2.15. The Pavia University scene similarly has no true outlier class, however, when analyzing the signatures it becomes apparent that classes such as trees, shadows, or painted metal sheets might be treated as the outlier class (where in the Pavia University image these have 3,064, 947, and 1,345 pixels, respectively). This scene, its ground truth, and a small subset of randomly sampled class signatures are shown in Figure 2.16. It is important to note, as depicted in these figures, the limitations of the given ground truth masks. In the Pavia University image, some labeled pixels, to include some background, have signatures that more resemble other classes. This may be due to sub-pixel traits and simply the complication of labeling each pixel for an image taken over such a large area. Figure 2.17 depicts two of the image's bands, representative of many of the bands, where certain pixels that are labeled as self-blocking bricks or background in the truth mask have much higher radiance values than their within-class counterparts. Further, when comparing the asphalt signatures between the two images, it can be seen from Figures 2.15 and 2.16 that the asphalt class behaves somewhat differently. This may be, in part, due to the apparent altitude difference. Specific class memberships are shown for each image in Table 2.3.

2.3.5 SpecTIR.

Three radiance data sets from SpecTIR Advanced Hyperspectral and Geospatial Solutions are used in this research [6]. Reflectance hypercubes are also available with CO_2 and Savitsky-Golay smoothing already applied, but such signatures have a different shape

Table 2.3: Pavia Sets Truth Data.

Class	Pavia Univ Number Pixels	Pavia Centre Number Pixels
Background	164624	635488
Asphalt	6631	3090
Meadows	18649	42826
Gravel	2099	
Trees	3064	7598
Painted Metal Sheets	1345	
Bare Soil	5029	2863
Bitumen	1330	6584
Self-Blocking Bricks	3682	2685
Shadows	947	7287
Water		65971
Tiles		9248

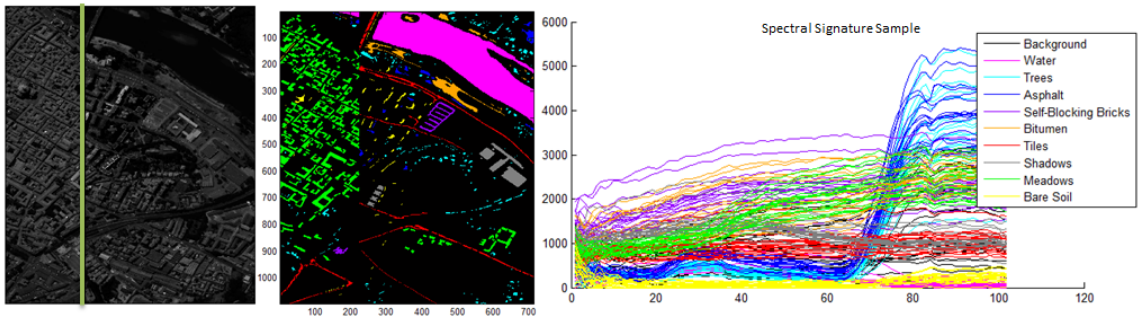


Figure 2.15: Pavia Centre Scene.

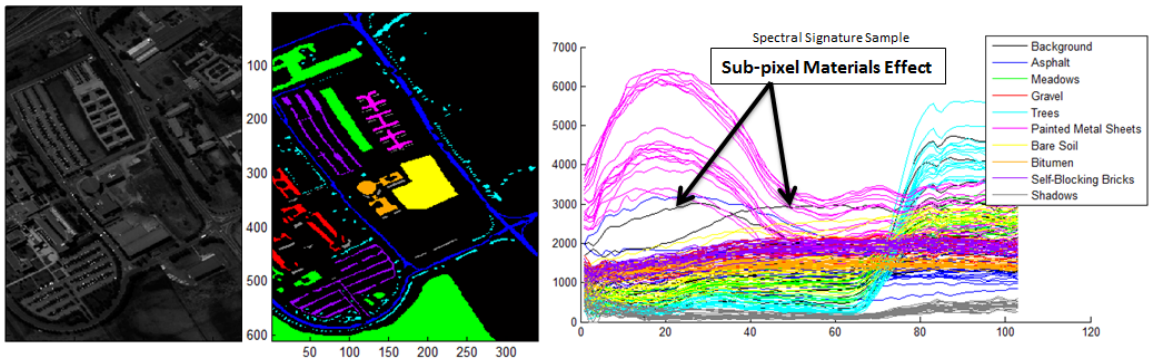


Figure 2.16: Pavia University Scene.

in comparison to those used for the HYDICE and AVIRIS images, and are also arguably less unique among different materials.

The first dataset is a 600 × 320-pixel urban and mixed environment image of Reno, NV with no associated truth mask. Values are over 356 spectral channels covering approximately 0.39-2.45 μm [6]. The second image also has no associated ground truth for objects or signatures, and was collected as a target of opportunity over the oil spill crisis in the Gulf of Mexico on June 6, 2010. The scene is 1160 × 320 pixels. The image is radiance collected at 2.2 m ground sample distance, over 360 spectral channels covering

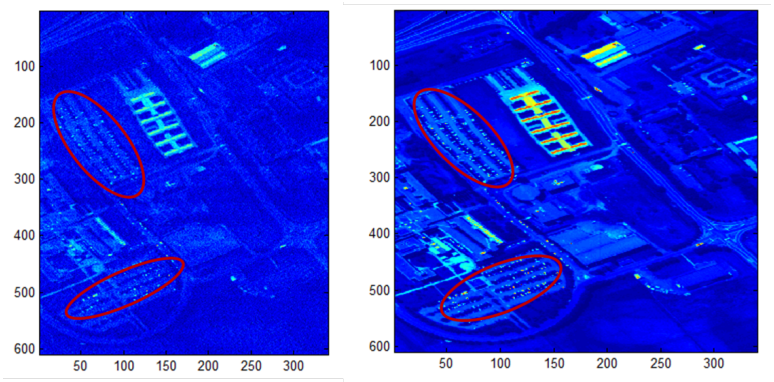
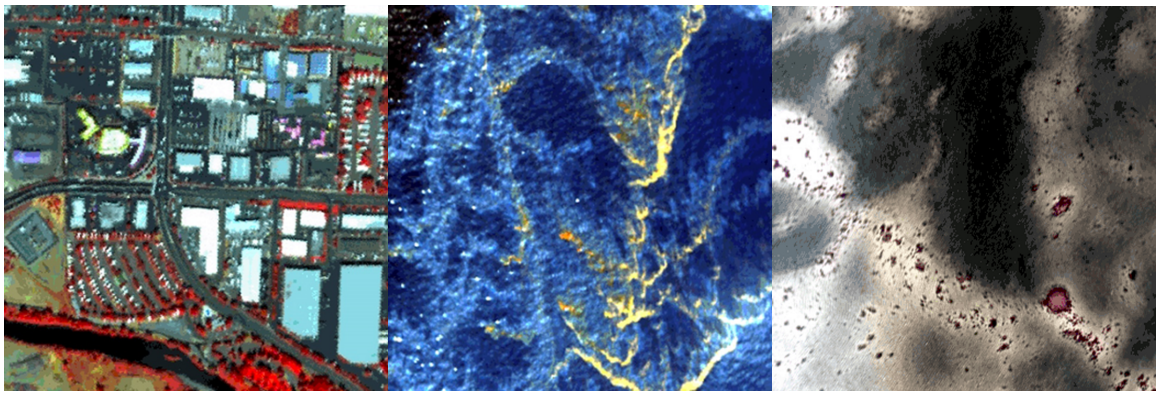


Figure 2.17: Pavia University Bands.

0.39-2.45 μm. The third image is an aquatic and coral reef sample over the Red Sea in Saudi Arabia, 600×960 pixels in size. Radiance values are over 128 spectral channels covering approximately 0.39-1 μm. The natural images are shown in Figure 2.18. Apparent objects, and/or crests of the spill can be seen in the Oil Spill image, while the coral reef can be seen in the Red Sea image.



(a) Reno. (b) Oil Spill. (c) Red Sea.

Figure 2.18: SpecTIR Images [6].

2.3.6 HyMap.

The HyMap sensor data sets were released for the Target Detection Blind Test project [194]. The scene is a 280×800-pixel image of Cooke City in Montana, USA over 126 bands with wavelengths from 0.453 to 2.496 μm and an approximate ground sampling distance of 3 m. Two data sets were provided: a self-test with truth/regions of interest of target placement and a blind test with no truth for target placement. Here,the self-test radiance dataset is used. Three vehicles types and four fabric panel colors with known signatures were used for targets. Thus, these pixels are ideal for a matching scenario, but here are also used as a reference for anomaly detection. Admittedly, this is limited in that the rural town is not a clean background and there may have been other vehicles present in addition to what was given as truth. The natural image and truth mask are shown in Figure 2.19. Defined regions of interest, noted here all as target pixels, include full-pixel, sub-pixel, and border pixels for a total of 145 target pixels. Similar to the HYDICE ARES images, these potential target pixels are further investigated in Section 4.3 to form the final truth data used in this research.

Next, in Chapter 3, general methods are discussed that recur throughout the remainder of the research. These methods include dimension reduction techniques, clustering techniques, and existing anomaly detection algorithms. Specific considerations for their application to the data sets presented in this chapter are discussed.

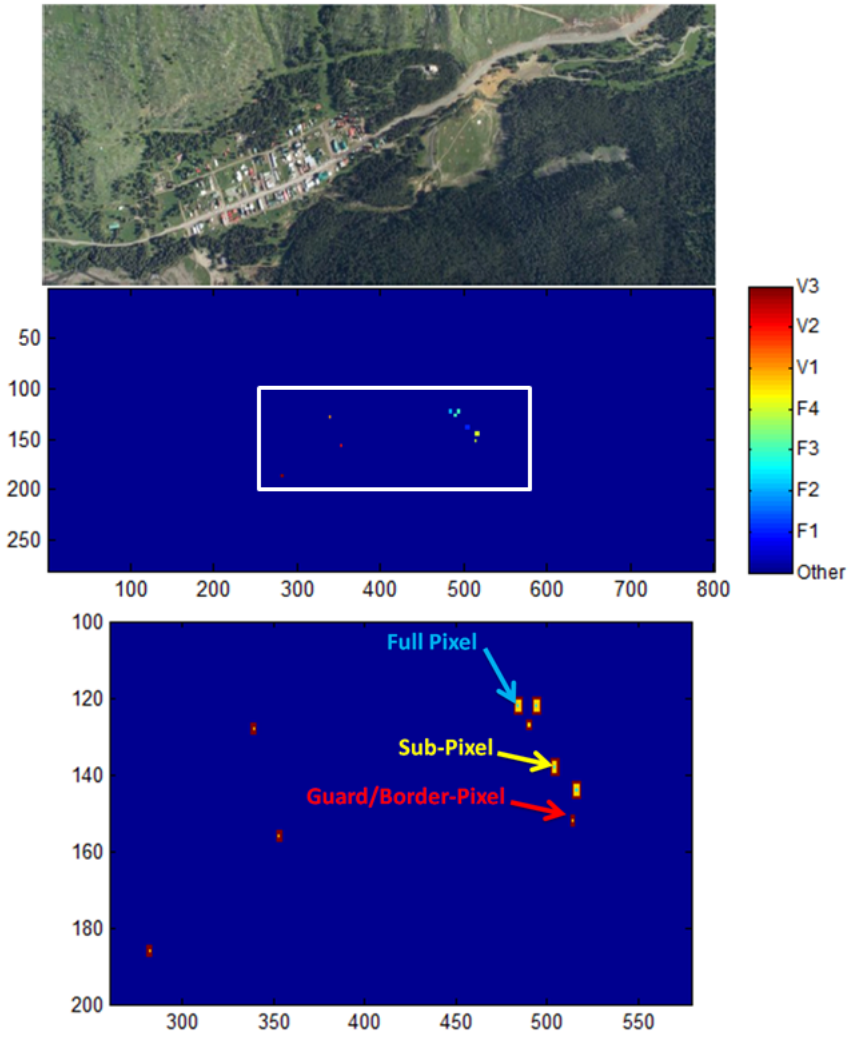


Figure 2.19: Cooke City, MT Image.

III. General Methods

For consistency of this document, some upfront notation is necessary. Let the hyperspectral data cube be denoted as a $m \times n \times p$ array with p spectral values for each of the $m \times n = N$ spatial locations, or pixels, of the image. For other multivariate data, let p denote the number of features and N the number of exemplars, where the data matrix is $N \times p$. C denotes a covariance matrix, and K a Gram matrix, unless otherwise stated. k denotes a number of centroids, neighbors, or exemplars being used within certain algorithms, or if it is used as a function, denotes the kernel function. $\mathbf{1}$ denotes a vector of ones.

3.1 General Dimension Reduction Techniques

Table 3.1 depicts properties of many dimension reduction techniques as taken from van der Maaten and van den Herik [149]. In the table, l denotes the number of local models in a mixture, d denotes the target dimension, i is the number of iterations, w is the number of weights, and r is the ratio of nonzero elements to total elements.

There are many linear and non-linear dimension reduction techniques, but Principal Component Analysis (PCA), Kernel PCA (KPCA), and Local Linear Embedding (LLE)-based techniques were explored in this research due to their accessibility and likeness to other methods. For instance, classical multi-dimensional scaling (MDS) using Euclidean distance for dissimilarity is related to PCA in that the MDS coordinates are the component scores from PCA [61]. Isomap, LLE, Laplacian Eigenmaps, and Maximum Variance Unfolding (MVU) can all be considered cases of KPCA using a specific kernel function due to their relation to the more general problem of learning eigenfunctions [149]. Some of these methods are related in that they involve building adjacency matrices based on nearest neighbors. LLE has shown great resemblance to MVU in the mappings produced, and diffusion maps with $t = 1$ are very similar to KPCA with a Gaussian kernel [149]. In fact,

Table 3.1: Dimension Reduction Technique Properties [149].

Technique	Convex	Parameters	Computational	Memory
Principal Component Analysis (PCA)	Y	None	$O(p^3)$	$O(p^2)$
Multi-Dimensional Scaling (MDS)	Y	None	$O(N^3)$	$O(N^2)$
Isomap	Y	k	$O(N^3)$	$O(N^2)$
Max Variance Unfolding	Y	k	$O((Nk)^3)$	$O((Nk)^3)$
Kernel PCA (KPCA)	Y	$kernel$	$O(N^3)$	$O(N^2)$
Diffusion Maps	Y	σ, t	$O(N^3)$	$O(N^2)$
Autoencoders	N	$netsize$	$O(iNw)$	$O(w)$
Local Linear Embedding (LLE)	Y	k	$O(rN^2)$	$O(rN^2)$
Laplacian Eigenmaps	Y	k, σ	$O(rN^2)$	$O(rN^2)$
Hessian LLE	Y	k	$O(rN^2)$	$O(rN^2)$
Local Tangent Space Analysis	Y	k	$O(rN^2)$	$O(rN^2)$
Locally Linear Coordination	N	l, k	$O(ild^3)$	$O(Nld)$
Manifold charting	N	l	$O(ild^3)$	$O(Nld)$

any technique that uses the eigen-pairs of a matrix of similarities or dissimilarities between exemplars can be related to KPCA.

Van der Maaten, Postma, and van den Herik [149] did a comparative study of many local and global dimension reduction techniques on a small variety of artificial and natural data sets, and found local techniques to suffer from issues due to large dimensionality, erroneous manifold assumptions, and the scale of eigenvalues complicating eigenproblems. Some global methods suffered similar issues, while the criticality of parameter choice such as with the correct kernel for KPCA, was highlighted.

3.2 Principal Component Analysis

Principal Component Analysis (PCA) generates a set of orthogonal vectors, any subset of which can be used to project into a subspace and where each vector accounts for some portion of the variance found in the data. Let $\hat{X}$ denote the centered data. Then the principal components are found by eigen-decomposing the covariance matrix $C = \frac{1}{N}\hat{X}^T\hat{X}$ as $C = V\Lambda V^T$, where Λ is the diagonal matrix of eigenvalues of C and V is the matrix of eigenvectors of C [68]. The eigenvector corresponding to the largest eigenvalue is the linear combination of original features that accounts for the most variance. Additionally, the eigenvector corresponding to λ_i accounts for the percentage $\frac{\lambda_i}{\sum_{i=1}^{p}\lambda_i}$ of the total variance found in the data. As a result of these properties and after sorting by eigenvalue magnitude, a number of leading eigenvectors are often chosen so as to account for some percentage of the total variance. The chosen eigenvectors, or components, are then a projection matrix W. Assuming some subset of the eigenvectors was chosen, this matrix can be used to approximately reconstruct the data, and squared residuals can be found for each exemplar using the row sums of the matrix,

$$\left(\hat{X} - \hat{X}WW^T\right) \circ \left(\hat{X} - \hat{X}WW^T\right), \tag{3.1}$$

where $\circ$ denotes the Hadamard product. The projection of the data onto the principal components, or set of scores, is simply $\hat{X}W$. The correlation of these scores with the original features yields a *loadings* matrix that represents the degree to which each component correlated with each feature [59].

In application to multi-spectral imagery, Green, et al. [80] investigated component Signal-to-Noise ratios of Airborne Thematic Mapper simulator data. They noted no definite trend relative to increasing noise with increasing component number once the components were ordered by eigenvalue magnitude. In order to generate ordered components in terms of image quality, they developed the maximum noise fraction (MNF) transformation, which is PCA-based but requires good estimates for the signal and noise covariance matrices.

Cheriyadat and Bruce [54] argued that PCA is not the optimal method for feature extraction in target detection applications. Specifically, they noted poor classifier performance on major components when within-class scatter dominated between-class scatter, where factors such as natural variation in the target material, environmental conditions, and sensor angle could cause such large within-class variance. Additionally, they argued that PCA may be poor in the multi-class case as local discriminatory statistics may be ignored. Their suggestions for alternatives required supervision, and they assumed that only major components (largest variance) were being used and that no additional techniques were applied to the component scores [54]. Their arguments raise valid concerns however, that are addressed beginning in Chapter 6. That is, how can good discriminatory components or mappings be selected, and how might information found from PCA and other techniques be fused such that those local discriminatory statistics are not lost?

3.3 Kernel Principal Component Analysis

One non-linear form of PCA uses kernels to perform standard PCA in a higher-dimension feature space $\mathcal{F}$. This enables a similar process for data with a non-linear structure and is referred to as Kernel PCA (KPCA) [97]. Assume some non-linear map

$\Phi : \mathbb{R}^p \to \mathbb{R}^d$, where $d > p$. This mapping sends the original data into an arbitrarily large, possibly infinite dimensional space. In this space, each centered exemplar is defined as,

$$\hat{\Phi}\left(\boldsymbol{x}_j\right) = \Phi\left(\boldsymbol{x}_j\right) - \frac{1}{N}\sum_{i=1}^{N}\Phi\left(\boldsymbol{x}_i\right), \tag{3.2}$$

for all j. The covariance matrix is then also,

$$C = \frac{1}{N}\sum_{j=1}^{N}\hat{\Phi}\left(\boldsymbol{x}_j\right)\hat{\Phi}\left(\boldsymbol{x}_j\right)^T. \tag{3.3}$$

To perform linear PCA in this space, eigenvalues λ and eigenvectors $\boldsymbol{v}$ are solved for in the higher dimensional space using the eigenproblem $\lambda\boldsymbol{v} = C\boldsymbol{v}$ [97, 200]. This implies that for any exemplar $\boldsymbol{x}_k \in \mathbb{R}^p$,

$$\lambda\left(\hat{\Phi}(\boldsymbol{x}_k)\cdot\boldsymbol{v}\right) = \hat{\Phi}(\boldsymbol{x}_k)\cdot C\boldsymbol{v}. \tag{3.4}$$

Each eigenvector $\boldsymbol{v}$ is a linear combination of the training exemplars in the new centered space,

$$\boldsymbol{v} = \sum_{j=1}^{N}\alpha_j\hat{\Phi}(\boldsymbol{x}_j). \tag{3.5}$$

Using Equations 3.3 and 3.5 in 3.4 yields for every $k = 1, \ldots, N$,

$$\lambda\sum_j \alpha_j\hat{\Phi}(\boldsymbol{x}_k)\cdot\hat{\Phi}(\boldsymbol{x}_j) = \frac{1}{N}\sum_j\alpha_j\sum_i\{\hat{\Phi}(\boldsymbol{x}_k)\cdot\hat{\Phi}(\boldsymbol{x}_i)\}\{\hat{\Phi}(\boldsymbol{x}_i)\cdot\hat{\Phi}(\boldsymbol{x}_j)\}. \tag{3.6}$$

But defining $\hat{K}_{ij} = \hat{\Phi}(\boldsymbol{x}_i)\cdot\hat{\Phi}(\boldsymbol{x}_j)$ and $\boldsymbol{\alpha} = (\alpha_1 \ldots \alpha_N)^T$, this simplifies to,

$$\lambda\boldsymbol{\alpha} = \frac{1}{N}\hat{K}\boldsymbol{\alpha} \to (N\lambda)\,\boldsymbol{\alpha} = \hat{K}\boldsymbol{\alpha}. \tag{3.7}$$

Therefore, if the dot products can be found, the eigenvalues $N\lambda$ and eigenvectors $\boldsymbol{\alpha}$ can be derived directly from $\hat{K}$ and Φ does not need to be known. In fact, the dot products are found using a kernel function. $\hat{K}$ is the modified, or centered form of the Gram matrix K, where $K_{ij} = \left\langle\Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j)\right\rangle = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for some kernel function k. Recalling that an entry in $\hat{K}$ is just the dot product of two vectors as found in Equation 3.2, by algebraic reduction it can be shown that $\hat{K}$ can be found from K as,

$$\hat{K} = K - \mathbf{1}_N K - K\mathbf{1}_N + \mathbf{1}_N K\mathbf{1}_N, \tag{3.8}$$

where $\mathbf{1}_N$ is a $N \times N$ matrix with values $1/N$ [200]. This subtracts the column means and the row means, and adds back the overall mean.

Because $\mathcal{F}$ is higher-dimensional than the original data, only non-zero eigenvalues should be considered. In order to normalize $\boldsymbol{\alpha}$, non-zero $\boldsymbol{v}$ are required to be normalized as $v^{(l)} \cdot v^{(l)} = 1$. Using equation 3.5 in this equation simplifies to the normalized coefficients,

$$\hat{\alpha}^{(l)} = \frac{\alpha^{(l)}}{\sqrt{(N\lambda_l)}}. \tag{3.9}$$

To clarify, only $\hat{\boldsymbol{\alpha}}$ and $\hat{K}$ are needed to project onto $\boldsymbol{v}$. This can be shown easily by considering that for some test data $\boldsymbol{y}$,

$$v^{(l)} \cdot \hat{\Phi}(\boldsymbol{y}) = \sum_{i=1}^{N} \hat{\alpha}_i^{(l)} \left(\hat{\Phi}(\boldsymbol{x}_i) \cdot \hat{\Phi}(\boldsymbol{y})\right) = \sum_{i=1}^{N} \hat{\alpha}_i^{(l)} \hat{K}^{\mathbf{test}}(\boldsymbol{x}_i, \boldsymbol{y}), \tag{3.10}$$

where $\hat{K}_{ij}^{\mathbf{test}}$ is the similarly centered form of $k(\boldsymbol{y}_i, \boldsymbol{x}_j)$. This is performed as,

$$\hat{K}^{test} = K^{test} - \mathbf{1}_N^M K - K^{test}\mathbf{1}_N + \mathbf{1}_N^M K \mathbf{1}_N, \tag{3.11}$$

where $\mathbf{1}_N^M$ is $M \times N$ with all entries $1/N$, for a test set with M exemplars [200].

One lingering question is what constitutes a kernel. To define this, Mercer's theorem is used [161]. Mercer's theorem states that a symmetric function $k(\boldsymbol{x}, \boldsymbol{y})$ can be expressed as an inner product,

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle \tag{3.12}$$

for some Φ if and only if $k(\boldsymbol{x}, \boldsymbol{y})$ is positive semidefinite,

$$\int k(\boldsymbol{x}, \boldsymbol{y}) g(\boldsymbol{x}) g(\boldsymbol{y}) dx dy \geq 0, \forall g \in \mathbb{L}_2 \tag{3.13}$$

or equivalently,

$$\begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & k(\boldsymbol{x}_1, \boldsymbol{x}_2) & \dots \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & \ddots & \\ \vdots & & \end{bmatrix} \tag{3.14}$$

is positive semi-definite for any collection $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ of exemplars [113]. That is, the Gram matrix is positive semi-definite for the set of data.

Some commonly used Mercer kernels include,

1. Polynomial: $(\boldsymbol{x} \cdot \boldsymbol{y} + c)^d$, where $c \in \mathbb{R}$, $d \in \mathbb{N}$,
2. Sigmoid: $\tanh(\boldsymbol{x} \cdot \boldsymbol{y} + c)$, where $c \in \mathbb{R}$,
3. Inverse Multiquadric: $\dfrac{1}{\sqrt{||\boldsymbol{x} - \boldsymbol{y}||^2 + \sigma^2}}$, for a given norm,
4. and the Gaussian/Radial Basis Function: $\exp\left(-\dfrac{||\boldsymbol{x} - \boldsymbol{y}||^2}{2\sigma^2}\right)$,

using the L_2 norm where c is some real constant, $d \in \mathbb{N}$ is the power, and $\sigma > 0$ is the spread parameter [146]. Within the context of KPCA, use of the Gaussian kernel only makes the Normality assumption in the higher-order space, and not the originating space. The dot product $\boldsymbol{x} \cdot \boldsymbol{y}$ is also a kernel, referred to as the linear kernel, but is closely tied to normal PCA. If $(\lambda_i, \boldsymbol{v}_i)$ are an eigen-pair for $X^T X$, then λ_i and $\boldsymbol{u}_i = \lambda_i^{-1/2} X v_i$ are an eigen-pair for XX^T. Doing PCA in this latter manner, so that scores are computed for variables rather than exemplars, has also been denoted as kernel eigenfaces when doing facial recognition assuming alignment of certain features across the images [224]. Typically with kernel eigenfaces, rather than computing a score for every pixel of an image, each image is reshaped to be treated as a column of pixel values, and thus, the transpose causes each column to be treated as an exemplar. Additionally, kernel eigenfaces differs from strict KPCA in that projected exemplars are often compared against predefined face classes for purposes of classification [162, 206]. Paiva, Xu, and Principe [170] showed that KPCA with a Gaussian kernel provided optimum entropy projections in the input space.

Li, et al. [138] introduced a similar concept for feature extraction of images. Motivated by a Matrix norm, they proved the Gaussian function,

$$k(X,Y) = \exp\left(-\frac{\sum_{j=1}^{n_1}\left(\sum_{i=1}^{n_2}\left(x_{ij}-y_{ij}\right)^2\right)^{1/2}}{2\sigma^2}\right), \tag{3.15}$$

to be a kernel function for images X and Y. In this manner, a series of images (or bands) could be reshaped to vectors in order to compute the kernel values and to perform KPCA on the images (or bands) rather than the pixels.

Bengio, Vincent, and Paiement [26] made an additional, interesting observation relative to KPCA. Based on learning the eigenfunctions of a kernel and the corresponding relation to KPCA, they noted that the KPCA embedding attempts to preserve the largest dot products in the feature space in the mean-squared error sense, *e.g.*, colinear exemplars.

3.4 Factor Analysis

Whereas PCA seeks a lower-dimensional representation that accounts for the variance of the features, factor analysis seeks a lower-dimensional representation that accounts for the correlations among features [68]. Thus, correlated features can be represented with a smaller set of new unobserved features, called factors. Specifically, a factor analysis model is,

$$\hat{X} = LF + E, \tag{3.16}$$

of p observable features (variables) $\hat{X} = \left[\hat{\boldsymbol{x}}_1 ... \hat{\boldsymbol{x}}_p\right]^T$, assumed zero-mean with finite variance, as linear combinations of n common factors $F = [\boldsymbol{f}_1 ... \boldsymbol{f}_n]^T$, plus uncorrelated noise or error terms $E = \left[\boldsymbol{e}_1 ... \boldsymbol{e}_p\right]^T$ [36]. These error components are zero-mean and mutually uncorrelated, and are additional sources of variation. The factor loading l_{ij} in the matrix L shows the degree to which feature i correlates with factor j, where high magnitude loadings for a factor reveal the contributing features.

In a way, PCA can be viewed as a similar model where no noise/error, *i.e.*, perfect data, is assumed. For feature X_i, $Var(X_i) = l_{i1}^2 + ... + l_{in}^2 + \psi_i$. The portion of the variance of the i-th variable contributed by the n common factors, $l_{i1}^2 + ... + l_{in}^2$, is then referred to as the communality, where ψ_i is the specific variance [218].

To estimate the factors initially, several methods exist. Principal component-based factor analysis uses the eigen-decomposition of C to estimate the factor loadings, where here C typically denotes the correlation matrix, but can alternatively be the covariance [59]. Let Λ denote the diagonal matrix of eigenvalues of C, and V the matrix of corresponding eigenvectors. Then $C = V\Lambda V^T$, as with PCA. The estimated factor loadings are given by

$$\hat{L} = \left[\sqrt{\lambda_1}\boldsymbol{v}_1, ..., \sqrt{\lambda_p}\boldsymbol{v}_p \right]. \tag{3.17}$$

The estimated factor scores can be found numerous ways, most commonly through unweighted least squares, maximum likelihood, or weighted least squares where the specific variances are used to weight the solution. In the case of the unweighted least squares solution, the factor scores are estimated by $\left(\hat{L}^T\hat{L}\right)^{-1}\hat{L}^T\hat{X}$ [218]. Thus, to estimate the specific variances, the following is used:

$$\Psi = C - \hat{L}\hat{L}^T. \tag{3.18}$$

The maximum likelihood method operates under the assumption that the common factors and error terms are multivariate normal. Under this assumption, $\hat{X} \sim \mathcal{N}\left(0, \Psi + \hat{L}^T\hat{L}\right)$ and the log-likelihood function to optimize is

$$-\frac{Np}{2}\log 2\pi - \frac{N}{2}\log|\Psi + \hat{L}^T\hat{L}| - \frac{N}{2}\text{tr}\left(\left(\Psi + \hat{L}^T\hat{L}\right)^{-1}C\right), \tag{3.19}$$

where $\text{tr}(A)$ denotes the trace of matrix A [186].

The unweighted least squares technique has the advantage of not requiring iteration to solve, and is thus more efficient. For this reason, and because of desirable results, it is used in this research. If using the covariance matrix, variance values are still just a sum of the

squared loadings and specific variance, as in Equation 3.18. In order to place the specific variance on a [0, 1] scale in this case, it can be re-scaled by dividing by the variance as

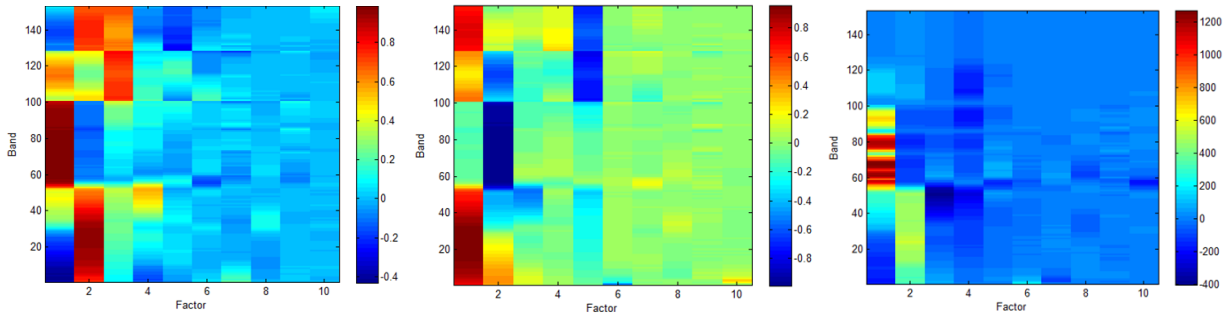
$$\psi_i = \left(\sum_{j=1}^{n} l_{ij}^2 \right) / Var(X_i). \tag{3.20}$$

Once the factors and loadings are estimated, rotation can be applied to change the coordinate system, and thus the loadings and scores. However, this rotation does not affect the feature structure [186]. Varimax is a popular rotation that rotates the factors orthogonally to maximize the variance of the squared loadings of a factor on the features. Such a rotation produces primarily large or small loadings for any feature, making interpretation of the factors more meaningful. In this research, this rotation is used in order to provide the easiest interpretation and to group the features. Other orthogonal and oblique rotations, where in the latter case, factors may be correlated, also exist [109, 151].

Although it is much more common to perform factor analysis using the standardized data and the correlation matrix, in this research the centered data and covariance matrix are used. If the data features are on the same scale with common variance, the two methods are equivalent. Here, the HSI and Arcene data have features on a common scale. Although the features do not have equal sample variance, the author found that not standardizing the data provided nearly the same or sometimes better discrimination for anomaly detection and feature selection on these data sets. For the HSI data, this is due in part to large numbers of bands being highly correlated. Using the covariance adds more discrimination amongst band coefficients.

Consider the ARES1F image as an example. Figure 3.1 shows the loadings matrices for three sets of factors using a model of $k = 10$ factors. The unweighted and MLE methods provide some similar factors for the correlation, while the covariance provides more discrimination between bands. This implies that the covariance-based factor scores truly use better subsets of bands to generate the scores, rather than just entire regions of the EM spectrum as the correlation factors are sometimes subject to doing (*i.e.*, the areas

of the pixel signatures between absorption locations). To further exemplify this, scores for various factors maps are shown in Figure 3.2. The first factor is similar across methods. The unweighted method on the correlation matrix provided no good discriminating factor, and so its best discriminating factor, F2, is shown for comparison in Figure 3.2(e). Meanwhile, the unweighted method on the covariance matrix provided a better discriminating factor for many of the targets than its MLE counterpart on the correlation matrix, in this case using Factor 5 as an example. As shown, certain background pixels appear anomalous in the MLE case.

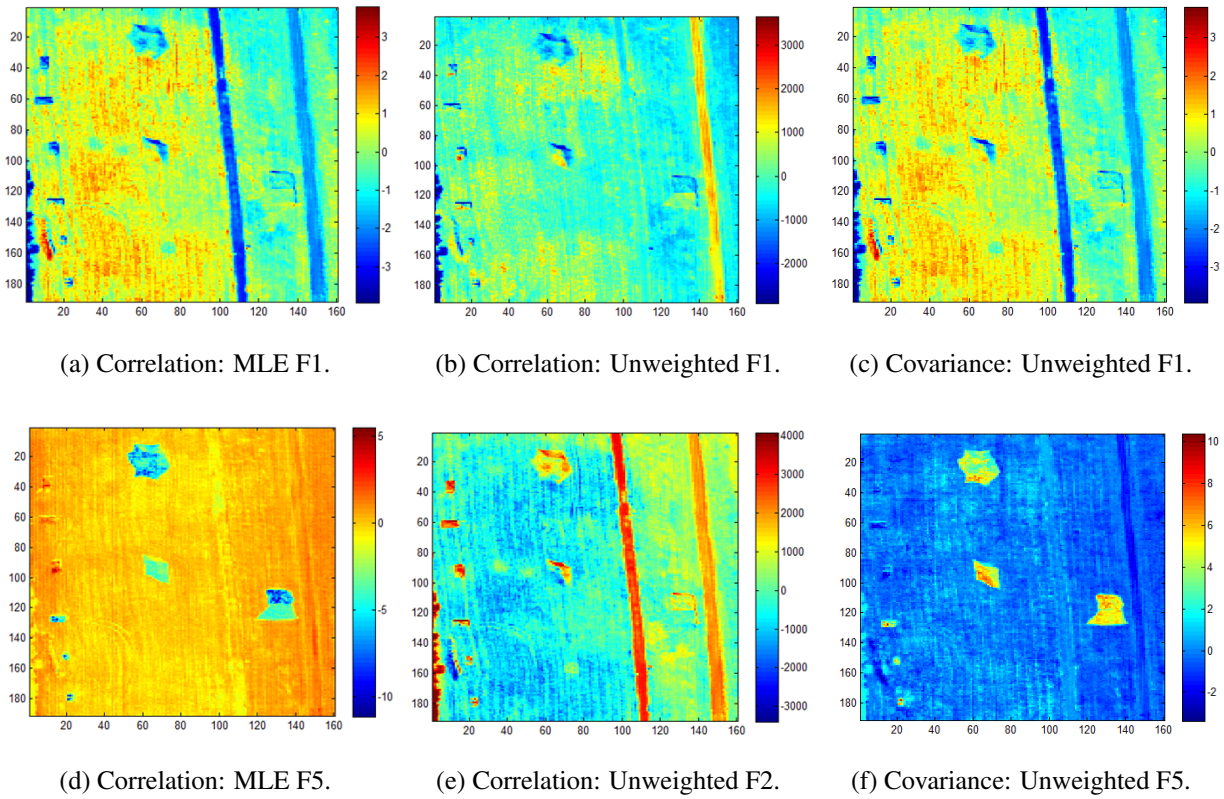


(a) Correlation: MLE. (b) Correlation: Unweighted. (c) Covariance: Unweighted.

Figure 3.1: Loadings Comparison.

3.5 Locally Linear Embedding

Locally Linear Embedding (LLE) is a dimensionality reduction technique developed by Roweis and Saul [181, 182]. The LLE algorithm assumes that each data exemplar is sampled from some underlying manifold, that itself is sampled sufficiently, such that each exemplar and its neighbors lies on or close to a locally linear patch of the manifold. The corresponding local geometry is characterized using linear coefficients that reconstruct each data point from its neighbors. A new neighborhood-preserving embedding is then



(a) Correlation: MLE F1. (b) Correlation: Unweighted F1. (c) Covariance: Unweighted F1.

(d) Correlation: MLE F5. (e) Correlation: Unweighted F2. (f) Covariance: Unweighted F5.

Figure 3.2: Factor Score Comparisons.

computed using eigenvectors related to these coefficients. The algorithm in full is shown as Algorithm 3.1. LLE allows for a non-linear reduction based on the geometry of the neighbors, whereas a method like PCA is restricted to linear directions. The resulting embedding is optimal in that the final eigen-problem equivalently optimizes $\sum_i |\boldsymbol{Y}_i - \sum_j W_{ij}\boldsymbol{Y}_j|^2$, where i is the current exemplar, $\boldsymbol{Y}_i$ is the embedding, W is the matrix of reconstruction coefficients, and j is a neighbor of i [182]. That is, using the same reconstruction coefficients found in the original space, the reconstruction of an exemplar from its neighbors is optimized in the new embedding. Of course, definition of the neighborhoods can largely affect embeddings found. Figure 3.3 depicts a one-dimensional embedding for the Banana dataset for two values of k, and their corresponding

neighborhood determinations in the original dataset. This is not necessarily the best example for a manifold method, as one class lies somewhat within the other, but the effect of varying k can be seen on the neighborhoods and the resulting embedding.

Algorithm 3.1 Locally Linear Embedding [181, 182]

1: Find a set of neighbors U_i for each exemplar $\boldsymbol{x}_i$,$i = 1, \ldots, N$, where $\boldsymbol{x}_i$ is not a neighbor of itself. The simplest means to do so is to use k-nearest neighbors.

2: *Solve for reconstruction weights:*

3: **for** i=1:N **do**

4: $\forall \boldsymbol{x}_j \in U_i$, let $\boldsymbol{z}_i = \boldsymbol{x}_j - \boldsymbol{x}_i$.

5: Compute the local covariance $C = ZZ^T$.

6: Solve for the column vector $\boldsymbol{w}$ of local weights by solving $C\boldsymbol{w} = \mathbf{1}$.

7: Normalize $\boldsymbol{w}$ such that $\sum_{j=1}^{|U_i|} w_j = 1$. For each neighbor j, let W_{ij} equal the corresponding entry from $\boldsymbol{w}$. Remaining elements in the i-th row of W are set to 0 or the matrix is treated as sparse.

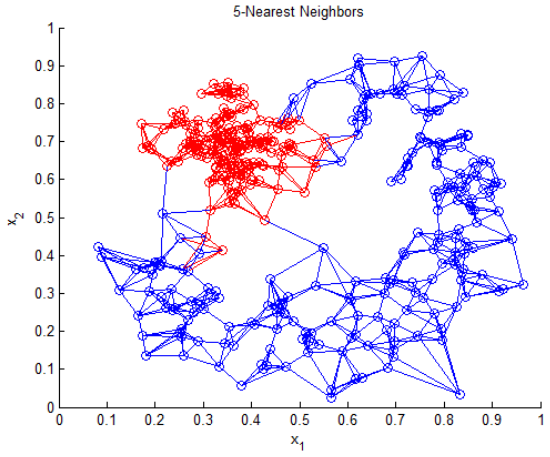
8: **end for**

9: *Solve for the embedding:*

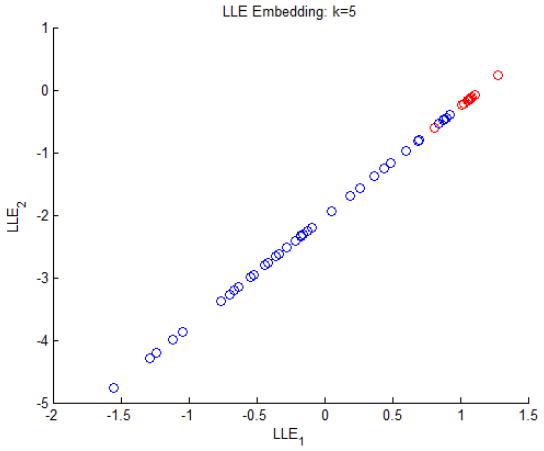
10: Create the sparse matrix $M = (I - W)^T(I - W)$.

11: Compute the eigenvectors of M and remove the eigenvector corresponding to the smallest eigenvalue (which has a value of zero and eigenvector of ones). Then the q-th dimension of embedded coordinates is the eigenvector corresponding to the q-th smallest eigenvalue.

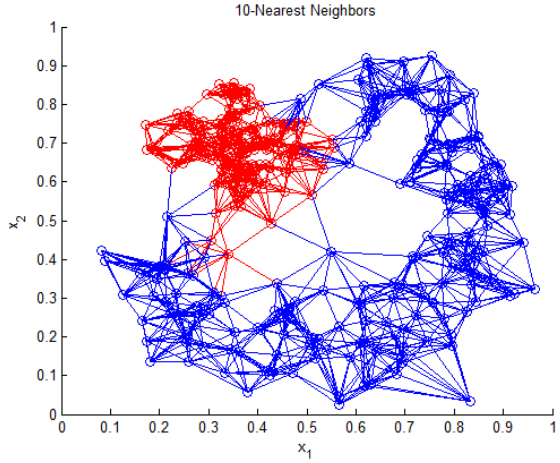
In reference to LLE and similar methods, van der Maaten, Postma, and van den Herik [149] noted the susceptibility of neighborhood graphs to outliers, over-fitting, and dimensionality issues. Additionally, they observed that LLE has a tendency to collapse large portions of data onto a single point if target dimensionality is too low. When the



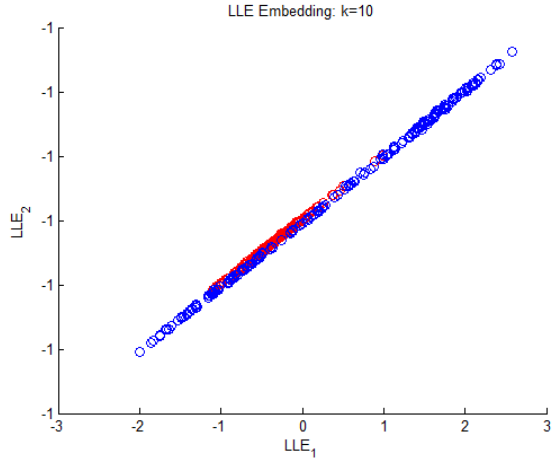
(a) 5-NNs.



(b) Embedding.



(c) 10-NNs.



(d) Embedding.

Figure 3.3: LLE Example for the Modified Banana Dataset.

number of neighbors, k is greater than the number of features p, the weight problem is ill-posed, and so a regularization term is used. The embedding results have been shown to be fairly insensitive to this term [187]. Note in Algorithm 3.1 that the smallest eigen-pair is ignored because the eigenvalue is zero and eigenvector is ones. It can be shown that this is always true by using the fact that $W\mathbf{1} = \mathbf{1}$, where this implies that $(I - W)\mathbf{1} = I\mathbf{1} - W\mathbf{1} = \mathbf{0}$, and so $(I - W)^T(I - W)\mathbf{1} = \mathbf{0}$.

LLE can be viewed as a special case of KPCA, as an eigen-problem is solved using a similarity matrix for the exemplars. However, unlike conventional KPCA, new test data cannot be immediately embedded into the new dimensions. Instead, new test data still has to be reconstructed from its neighborhood in both the original and new spaces. To solve this issue, He et al. [96] slightly modified the LLE algorithm. They instead used the smallest eigenvectors as projection vectors to yield the embedding, where they solved for the eigenvectors using the generalized eigenvector problem $X^T M X \boldsymbol{v} = \lambda X^T X \boldsymbol{v}$. This technique is used in this research. Here, X is the data matrix, M is as defined in Algorithm 3.1, and λ and $\boldsymbol{v}$ are the eigenvalue and eigenvector. Chen, Qu, and Lin [53] instead used a Generalized Regression Neural Network to learn the mapping of the training data in a supervised sense. This enabled them to embed the test data using the network. Kouropteva, Okun, and Pietikäinen [128] presented three ways to handle test exemplars. Two techniques were based on linear generalizations, as the main assumption in the method is that the manifold is locally linear. Letting X^{N+1} be the matrix of k-nearest neighbors for test point $\boldsymbol{x}_{N+1}$ and Y^{N+1} be the embedding of those nearest neighbors, then the equation $Y^{N+1} = X^{N+1} Z$ would be approximately true, where Z is an unknown linear transformation matrix that can be solved for by least squares. Z could then be applied to the test point. Alternatively, the reconstruction weight vector for $\boldsymbol{x}_{N+1}$ was applied directly to the embeddings of the nearest neighbors to yield its embedding. The third technique they proposed was to update M only as needed due to changes in nearest neighbors with the inclusion of a new test point. Denoting Y as the embedding, they assumed minimal change to the eigenvalues with the inclusion of very few new test points at a time, and so treated the eigenvalues as constants in order to avoid resolving the eigen-problem. This enabled them to solve $YMY^T = \Lambda$ for the new embedding of a few test points at a time.

LLE can be performed using any neighborhood determination, not just by using k-nearest neighbors. A simple radius can also be used where for an exemplar $\boldsymbol{x}_i$, $\boldsymbol{x}_j$

belongs to its neighborhood if $\|\boldsymbol{x}_j - \boldsymbol{x}_i\| < \delta$, where $\delta > 0$ [147]. Because a small change in k can provide very different neighborhood structures, Lu [147] noted that 2- or 3-nearest neighbors are often used for robustness with high-dimensional data, but that this was still relatively random as was any choice for δ for the radius method. Due to interest in also reducing sensitivity to noise in the data, and its effect on the resulting nearest neighbor determinations, Lu [147] developed a Robust LLE (RLLE) algorithm. In order to handle error caused by noise, three aspects were derived that need to be compromised: 1) depression of noise (implying larger neighborhoods), 2) maximizing the smallest eigenvalue (implying smaller neighborhoods), and 3) reducing the magnitude of the weights. In order to do this, a neighborhood ball was proposed for better intrinsic neighborhood determination as the change to the base LLE algorithm. For exemplar $\boldsymbol{x}_i$, let $r_i = \min_{j=1,\ldots,N, i\neq j} \left\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2\right\}$. Using Lu's neighborhood ball, $\boldsymbol{x}_j$ is in the neighborhood of $\boldsymbol{x}_i$ (a nearest neighbor) if,

$$\|\boldsymbol{x_i} - \boldsymbol{x}_j\|_2 \leq r_i + r_j. \tag{3.21}$$

Chen and Qian [50] improved computational speed of LLE on HSI by using a neighborhood window within which to find neighbors, reducing the problem size when finding k-nearest neighbors while also incorporating spatial information. However, they admitted that the size of the window could determine the success of the method. Ziemann, Messinger, and Albano [233] solved the k-nearest neighbor problem in LLE by using natural nearest neighbors. First, the 1-nearest neighbor is found for each point x_i. Then, $nb(i)$ is defined as the number of other points that exemplar i is the nearest neighbor for. Next, the 2-nearest neighbors are found for each point, and $nb(i)$ is updated to reflect how many points exemplar i is a nearest neighbor for. This is done iteratively until every point i has $nb(i) > 0$. The k-connectivity for each point i is then set as $nb(i)$, meaning points in dense regions use a large k and points in sparse regions use a small k.

A supervised form of LLE was developed by de Ridder et al. [179] so as to shape neighborhoods based on class information. They proposed enlarging the distance between two points of different classes by adding a penalty αD, where $0 \leq \alpha \leq 1$ and D is the maximum distance between any two points in the dataset. Zhang and Zhao [229] took this to a partially supervised case by proposing to multiply this term by $1 - P$, where P is the probability that the two exemplars are of the same class. RLLE, and a new combination of supervised and RLLE using $\alpha = 0.5$ for the Banana dataset are shown in Figure 3.4.

3.6 Discriminant Analysis

Per Duda, Hart, and Stork [68], whereas a method such as PCA seeks directions for representation, discriminant analysis seeks directions for discrimination. In particular, (Fisher's) linear discriminant analysis (LDA) seeks to maximize the distance between projected class means, while minimizing variances of projected classes. In order to optimize this measure of discrimination, the method must be supervised.

Consider a two-class problem. Let $S_B = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T$ be the between-class variance and $S_W = \sum_{i=1}^{2} \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$ be the within-class variance for the data. Then the criterion to optimize for a projection direction *bmw*, sometimes noted as the Rayleigh coefficient or quotient[159], is,

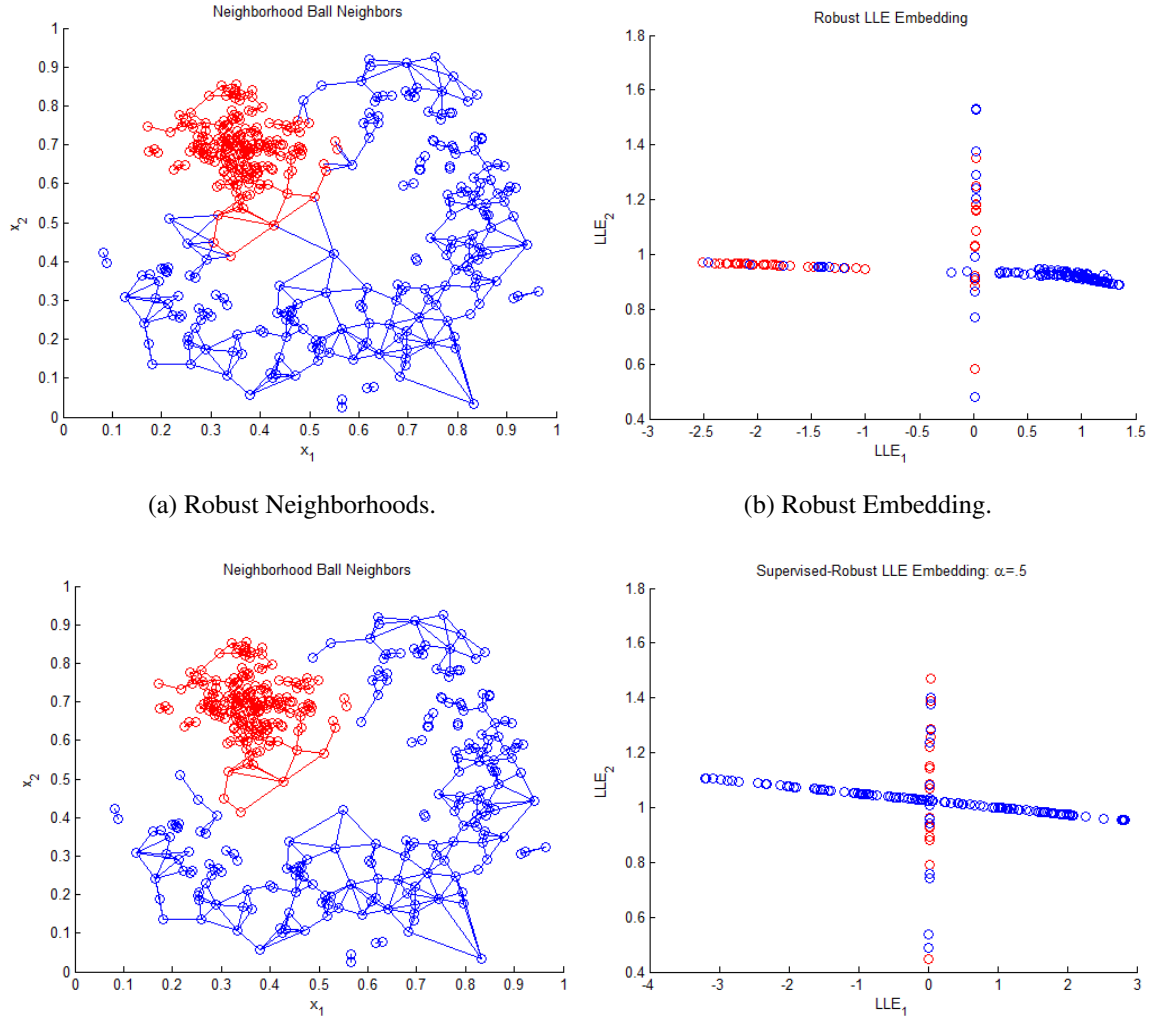
$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}. \tag{3.22}$$

Equivalently, this is the ratio of between-class variance over within-class variance for the projected data [68]. A vector $\boldsymbol{w}$ that maximizes this quotient satisfies,

$$S_B \boldsymbol{w} = \lambda S_W \boldsymbol{w}, \tag{3.23}$$

for some constant λ [68]. If S_W is nonsingular, then there is an eigenvalue problem,

$$S_W^{-1} S_B \boldsymbol{w} = \lambda \boldsymbol{w}. \tag{3.24}$$



(a) Robust Neighborhoods.

(b) Robust Embedding.

(c) Supervised-Robust Neighborhoods.

(d) Supervised-Robust Embedding.

Figure 3.4: Banana Dataset RLLE and Supervised RLLE Example.

For the normal, equal-covariance case, the optimal decision rule using $\boldsymbol{w}$ can be provided via a threshold [68].

With more than two classes, c, this process can be generalized to $c-1$ discriminant functions, often named Multiple Discriminant Analysis. Similarly to before, $S_W = \sum_{i=1}^{c} \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$. Now however, S_B is changed so that the total scatter found in

the data is $S_B + S_W$. This defines $S_B = \sum_{i=1}^{c} n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$, where $\boldsymbol{\mu}$ is the overall mean of the data and $n_i = |X_i|$ reflecting the size of each class. To generalize the Rayleigh coefficient, the criterion becomes the quotient of determinants relative to the matrix of optimal directions W,

$$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|}. \tag{3.25}$$

Each solution w_i still satisfies Equation 3.23, and so they are the eigenvectors of $S_W^{-1} S_B$ similar to before [68]. However, note these formulations assume equality of class covariance matrices.

Lu, Plataniotis, and Venetsanopoulos [146] noted the significance of the *small sample size* problem in LDA towards the eigenface application. That is, with increasing space dimensionality, one needs exponentially many patterns to sample the space properly to avoid high variance in the estimation of S_B and S_W. They noted PCA as a common pre-processing step, but commented that discarded components can contain significant discriminatory information. In this research, it is desirable to avoid this issue by building sufficiently large, yet representative skeletons and by evaluating the discriminatory nature of eigenvectors. Here, there are often many more exemplars than are typically found for facial recognition tasks.

Fukunaga and Mantock [77] proposed using nearest neighbors and associated weighting functions to yield a nonparametric form of discriminant analysis. Zhu and Hastie [230] proposed a nonparametric method based on a log-likelihood ratio relative to the Rayleigh coefficient. The Rayleigh coefficient can also be generalized to the kernel case, where the normality assumption would be made in the higher-order space and not the originating space. This is referred to as Kernel LDA, Kernel Fisherfaces, or Generalized Discriminant Analysis [146, 224]. Mika, et al. [159] first expanded this for the two-class case.

Recall, the Rayleigh coefficient from Equation 3.22, and let Φ be the non-linear mapping and N_i be the number of exemplars in class C_i. Then $\boldsymbol{w}$ in the higher-dimensional space is sought, where the scatter matrices are,

$$\begin{aligned} S_B &= (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T, \boldsymbol{m}_i = \frac{1}{N_i} \sum_{\boldsymbol{x} \in C_i} \Phi(\boldsymbol{x}), \\ S_W &= \sum_{i=1,2} \sum_{\boldsymbol{x} \in C_i} (\Phi(\boldsymbol{x}) - \boldsymbol{m}_i)(\Phi(\boldsymbol{x}) - \boldsymbol{m}_i)^T. \end{aligned} \tag{3.26}$$

Fortunately, $\boldsymbol{w}$ can be written as a linear combination of the mapped data $\boldsymbol{w} = \sum_{\boldsymbol{x} \in \mathcal{X}} \alpha_x \Phi(\boldsymbol{x})$. This yields, using a kernel to model the inner product,

$$\begin{aligned} \boldsymbol{w}^T \boldsymbol{m}_i &= \frac{1}{N_i} \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{z \in C_i} \alpha_x (\Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{z})) \\ &= \frac{1}{N_i} \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{z \in C_i} \alpha_x k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\alpha}^T \boldsymbol{\mu}_i \text{ where } \boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{\boldsymbol{x} \in C_i} K_x. \end{aligned} \tag{3.27}$$

Thus, the numerator of the Rayleigh coefficient becomes $\boldsymbol{\alpha}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T\alpha = \boldsymbol{\alpha}^T M \boldsymbol{\alpha}$. In a similar fashion, the denominator of the Rayleigh coefficient becomes $\boldsymbol{\alpha}^T \left(K(I - \boldsymbol{v}_1\boldsymbol{v}_1^T - \boldsymbol{v}_2\boldsymbol{v}_2^T)K^T\right)\boldsymbol{\alpha} = \boldsymbol{\alpha}^T T \boldsymbol{\alpha}$, where $(v_j)_i = 1/\sqrt{N_j}$ if exemplar i belongs to class j and 0 otherwise, and T is used to denote $K(I - \boldsymbol{v}_1\boldsymbol{v}_1^T - \boldsymbol{v}_2\boldsymbol{v}_2^T)K^T$ [159].

This gives the Rayleigh coefficient in terms of $\boldsymbol{\alpha}$ vice $\boldsymbol{w}$,

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T M \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T T \boldsymbol{\alpha}}. \tag{3.28}$$

The optimal leading eigenvector $\boldsymbol{\alpha}$ is then found from $T^{-1}M$, analogous to its linear counterpart. Projections onto the optimal $\boldsymbol{w}$ can be made using the kernel and $\boldsymbol{\alpha}$ by $\boldsymbol{w} \cdot \Phi(\boldsymbol{z}) = \sum_{\boldsymbol{x} \in \mathcal{X}} \alpha_x k(\boldsymbol{x}, \boldsymbol{z})$.

With small sample size, the kernel discriminant problem can be ill-posed with respect to inversion. To solve this Mika, et al. [159] proposed adding a multiple of the Identity matrix to T. Others have done PCA (linear or kernel) before doing the discriminant analysis [223].

3.7 Wavelets

Wavelets are a prevalent topic in the HSI literature and so are discussed here at some length for completeness. Originally, this research also included investigations using

wavelets. However, due to discovered flaws in those methodologies, that research is not included in this document. Wavelets have been used for de-noising of images as well as to generate another space for feature extraction, as wavelets can be considered a kernel in KPCA.

For two-dimensional images, moments can be used as pattern features. Specifically, the moment definition using a basis function or moment weighting kernel $\psi(x, y)$, and an image intensity function $f(x, y)$ is given as,

$$\Psi_{qr} = \int_x \int_y \psi_{qr}(x, y) f(x, y) dx dy, \; q, r = 0, 1, 2, \ldots. \tag{3.29}$$

Legendre and Tchebycheff moments, or more specifically, their discrete approximations, have been used to generate image features [163]. This moment-representation is mentioned here as it is similar to the method of wavelets, in its simplest form.

Let $\psi_{a,b}(x)$, $a \in \mathbb{R}\backslash\{0\}$, $b \in \mathbb{R}$ be a family of functions defined as translations and re-scales of a single function $\psi \in \mathbb{L}_2(\mathbb{R})$,

$$\psi_{a,b}(x) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{x - b}{a}\right). \tag{3.30}$$

The function ψ is called the *mother wavelet* and is assumed to satisfy the admissibility condition,

$$C_\psi = \int_{\mathbb{R}} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty, \tag{3.31}$$

where $\Psi(\omega)$ is the Fourier transform of $\psi(x)$. This implies that $\int \psi(x) dx = 0$ [208]. For any $\mathbb{L}_2$ function $f(x)$, the continuous wavelet transformation acting on f is defined as,

$$CWT_f(a, b) = \langle f, \psi_{a,b} \rangle = \int_{\mathbb{R}} f(x) \overline{\psi_{a,b}(x)} dx, \tag{3.32}$$

where a and b vary continuously. The *resolution of identity* relation (or inverse) is then,

$$f(x) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_0^{\infty} CWT_f(a, b) \, \psi_{a,b}(x) \frac{1}{a^2} da db. \tag{3.33}$$

To actually compute the continuous transformation efficiently, discrete values of a and b can be used for critical sampling, and multiresolution analysis (MRA) can be used [208]. Wavelets easily extend to higher dimensions, such as for the transformation of HSI data.

Discrete forms also exist for wavelet transformations, and are advantageous in that they are $O(N)$. Different discrete transformations exist, but one popular method is the cascade algorithm, which processes the image at different scales ranging from fine to coarse in a tree-like algorithm [208]. Discrete transforms can often be described as a set of inner products between a finite-length sequence and a discretized wavelet basis.

The Discrete Wavelet Transform (DWT) is often used for image compression, is effective for multi-resolution decomposition, and has been used to extract features for face recognition [162]. The DWT reduces image resolution, but maintains local information in space and frequency domains. Admittedly, full development of the DWT can be complex, and so a brief development follows. For full details, Vidakovic and Mallat [153, 208], among others, have dedicated many pages in books to the subject.

For a two-dimensional image, one approach to compute the transform is to use four different filters,

$$\begin{aligned} \phi(n_1, n_2) &= \phi(n_1)\phi(n_2), \\ \psi^H(n_1, n_2) &= \psi(n_1)\phi(n_2), \\ \psi^V(n_1, n_2) &= \phi(n_1)\psi(n_2), \\ \psi^D(n_1, n_2) &= \psi(n_1)\psi(n_2), \end{aligned} \tag{3.34}$$

where n_1 is the horizontal direction, n_2 the vertical direction, ϕ the scaling function which is essentially a low-pass, or averaging, filter, and ψ the wavelet function which is essentially a high-pass filter [162]. A low-pass filter allows low values, as determined by some criterion, to pass unchanged and reduces high values. Similarly, a high-pass filter allows high values to pass unchanged and reduces low values. The product $\phi(n_1, n_2) = \phi(n_1)\phi(n_2)$ is the application of the low-pass filter to the horizontal direction and the vertical direction, with similar meaning for the other products where the alphabetic exponent on the ψ filter

denotes the direction of the high-pass filter (where D for diagonal represents horizontal and vertical). Applying the four filters yields four sets of coefficients for a two-dimensional image.

The scaling function $\phi(n_1, n_2)$ represents an approximation to the image, $\psi^H(n_1, n_2)$ and $\psi^V(n_1, n_2)$ represent the respective changes of the image along the horizontal and vertical directions, and $\psi^D(n_1, n_2)$ represents the high frequency component of the image. These latter three sets are sometimes referred to as *detail* coefficients. To further decompose the image, the four filters can be re-applied to $\phi(n_1, n_2)$ [153]. An example of this is shown in Figure 3.5, where subscripts denote the level of resolution. In facial recognition tasks, the approximation images have been found to be the richest for finding common features with which to classify [108, 162].

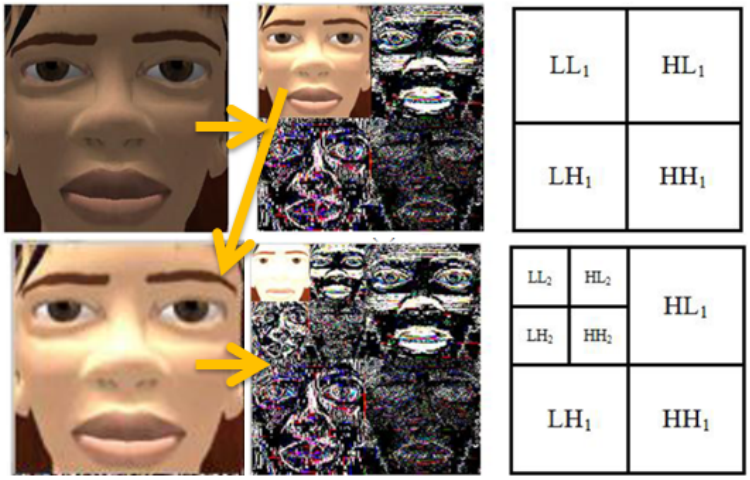


Figure 3.5: DWT Decomposition [162].

Wavelets have several nice properties. They generate local bases, and in many cases they are either compactly supported or they decay exponentially. This is nice in that the wavelets are most often orthogonal. Further, wavelets disbalance the energy in data while

preserving the total energy due to the orthogonality of the transformation. This means that the energy found in the original data is put into fewer coefficients in the wavelet coefficient space, *i.e.*, a more sparse or more concentrated representation [208]. Additionally, as a consequence of orthogonality, orthogonal wavelet transformations map white noise to white noise. Hence, correlated signals can become almost uncorrelated in the wavelet domain. On the contrary, wavelets provide sets of detail and approximation coefficients such that, although the wavelet coefficient space is more concentrated, it is also more complex in that there are sets of coefficients.

3.7.1 Shrinking/Smoothing.

Wavelets are popular, in part, because they can be easily used to de-noise an image by eliminating coefficients with a low magnitude. However, the exact means to do so without significantly changing properties of the image is not always trivial. The process itself is referred to as *shrinkage*, where the shrinkage function S is a non-decreasing function in terms of a coefficient's magnitude. Proper choice of shrinkage is very important, as it is easy to show that different scales in an image can provide entirely different representations of an object in a scene, but with the right choice of shrinkage, noise can be eliminated from the scene without over-smoothing [208].

The simplest form of shrinkage is thresholding and takes two forms, *soft* and *hard*. Hard-thresholding does not necessitate continuity of the shrinkage function and for coefficients c and threshold λ, it takes the form,

$$S\left(c,\lambda\right)=c\mathbf{1}\left(|c|>\lambda\right),\lambda\geq 0,c\in\mathbb{R}. \tag{3.35}$$

Soft-thresholding is continuous, and takes the form,

$$S\left(c,\lambda\right)=\left(c-\operatorname{sgn}(c)\cdot\lambda\right)\mathbf{1}\left(|c|>\lambda\right). \tag{3.36}$$

The choice of threshold can be made several ways. Risk can be used as one basis for this choice, where risk is the reconstruction error due to the shrinkage. If it is assumed that

the signals are a realization of a random vector F with added Gaussian white noise W of variance σ^2, *i.e.*, $X = f + W$, then a diagonal estimator of f in a basis $B = \{g_m\}_{0 \leq m < N}$ is:

$$\tilde{F} = DX = \sum_{m=0}^{N-1} a_m \left(X_B\left[m\right]\right) X_B\left[m\right] g_m. \tag{3.37}$$

In this representation, using a_m as the thresholding estimator, the risk of thresholding can be analyzed. Specifically, it can be shown that a threshold smaller than $\sigma\sqrt{2log_e N}$ reduces the risk associated with thresholding, although it is not optimal [153]. This is sometimes referred to as the universal threshold.

Mallat [208] noticed that for a variety of images and signals, the distributions of the wavelet coefficients were symmetric about zero and had a sharp peak at zero. Therefore, he modeled the distributions using the exponential power family and designed percentile-based thresholds. Several other methods exist, including cross-validation, block-thresholding, and Lorentz Curve thresholding. For the latter, the Lorentz curve for the distribution of the energy of the wavelet coefficients is used to define a threshold, where the Lorentz curve for a random variable X is defined as $L(q) = \frac{1}{\mu}\int_0^{\xi_q} xdF(x)$, where ξ_q is the population qth quantile. Defining $\hat{q}_0$ as the proportion at which the gain by thresholding an additional element is smaller than the loss in energy, the $\hat{q}_0 \cdot 100\%$ coefficients with smallest energy are replaced by zero [208]. Smooth shrinkage can also be used, where coefficients are set according to some smooth function.

3.7.2 Application of Wavelets to HSI.

For HSI specifically, wavelets have been used to de-noise prior to dimension reduction, and have also been used as a new space within which to perform dimension reduction [21, 39, 175]. Liu [143], Shen and Jia [189], and Xie et al. [221] used various mother wavelets for purposes of feature extraction and selection and multi-scale KPCA. In the cases where they used wavelets to de-noise images, noise had been artificially added to the image. Baghbidi et al. [20] performed variations of the RX detector on DWT coefficients, and showed benefit vice the local RX detector. Gupta and Jacobson [87] proved the

equivalence of PCA and PCA done on unscaled wavelet coefficients, as well as related eigenspectra of smoothed wavelet coefficients. These equivalences are depicted in Figure 3.6.

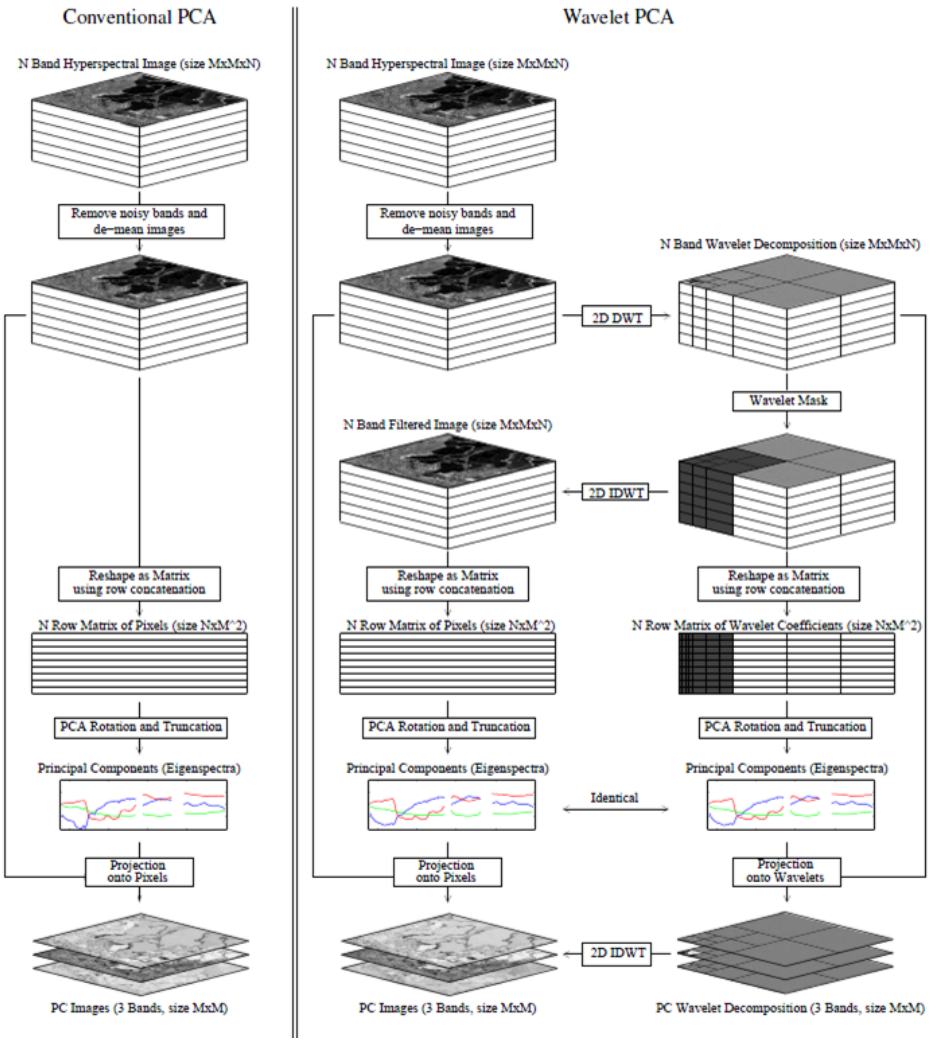


Figure 3.6: Workflow Diagrams for PCA and PCA Involving Wavelets [87].

In this research, wavelets are not used for de-noising. The factor analysis method constructed in Chapter 4 is designed specifically to remove the noisy bands. Although this may leave some noise from the sensor in the image, in practice using 'optimal' shrinkage

methods from Section 3.7.1 on the resulting images did not seem to alter characteristics of the images in a meaningful way. Higher levels of smoothing, meanwhile, appeared to overly alter the pixel signatures. Wavelets are also not used as an input for dimension reduction in this research as the presence of sets for the wavelet coefficients, even in the MRA case, did not truly reduce dimension in a desirable enough way for the purposes here.

3.8 k-Nearest Neighbors

The concept of k-nearest neighbors is simple in that the k-nearest exemplars according to some distance or similarity metric are sought for each exemplar in the dataset. However, as the size of the dataset grows, so too does the computational expense of computing the distances and determining the nearest neighbors. As this algorithm is often used for neighborhood determination within techniques such as LLE and others explored in this research, such as the visualization presented in Section 4.2, it is important to briefly discuss speed improvements to the basic k-nearest neighbor algorithm.

The k-d tree structure splits nodes into subtrees based on a value for the i-th coordinate. For example, consider Figure 3.7. The first coordinate is used to split the data, with 3 being the split value. Next, each subtree is split. In the case of the left subtree, the data is split at the median value of the second coordinate, 4. For the right subtree, there are only two datapoints, and therefore one of the values is chosen to split the subtree. This yields a tree structure for the datapoints that enables quicker searching vice building a full distance matrix. The capability to generate such a tree is available within Matlab®, but search performance still degrades exponentially with increasing dimensionality [166].

Nene and Nayar [166] proposed a simple algorithm to find neighbors by slicing each dimension, one at a time, keeping points within some constant ϵ of the current point. However, this algorithm does not guarantee k neighbors. Approximate nearest-neighbor algorithms attempt better performance by only guaranteeing that distances used

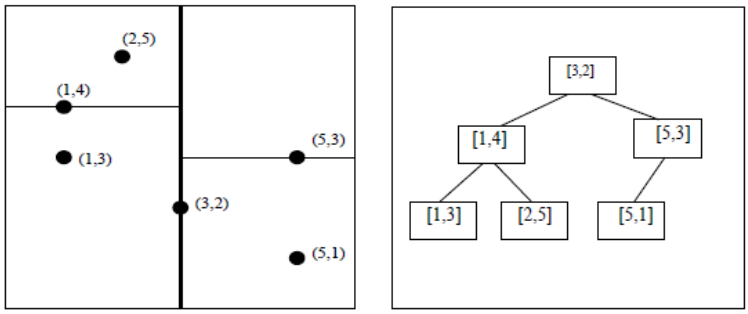


Figure 3.7: KD Tree Example [12].

are accurate to within some ϵ or factor of the true distances [16, 201]. Hwang and Wen [102] did a partial distance search in the discrete wavelet domain to reduce the number of features for the initial neighborhood determinations. Cheng, Fang, and Saad [51] used divide and conquer approaches to approximate neighbors. For purposes of this research and after some testing, only the kd-tree is used here to improve the *k*-nearest neighbor algorithm (mentioned primarily for purposes of run-time analysis). Additionally, the author found that chunking the data, in the case of large data sets, improved efficiency. That is, computing the nearest neighbors in the data sets for only 1,000 or a few thousand exemplars at a time, in conjunction with vector operations in Matlab®, vastly improved computational efficiency.

3.9 Clustering

A few prevalent clustering algorithms are presented in this section, as is some initial analysis for *k*-means. There are many issues associated with clustering, especially for purposes of this research. Some of these issues are introduced here, but further evaluation is presented in Section 7.3

3.9.1 k-Means.

k-means is perhaps the most common clustering algorithm. In *k*-means, a number of clusters *k* is chosen in advance. Given a similarity metric, each exemplar is assigned to its

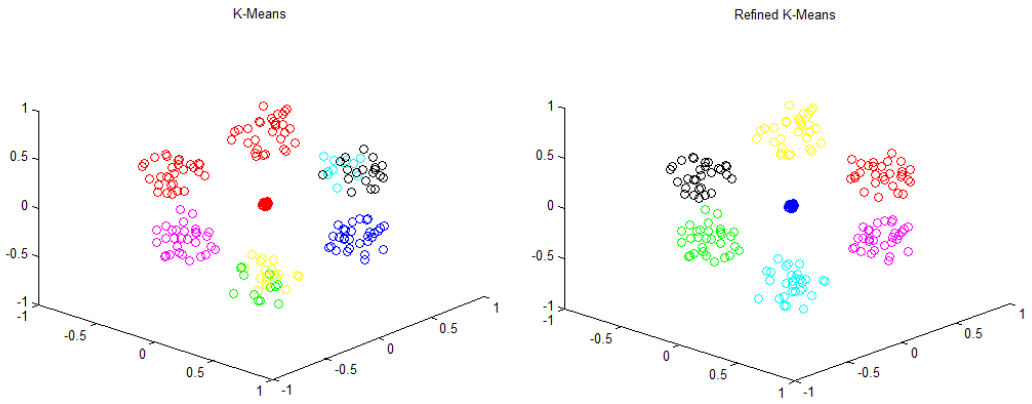
closest cluster as determined by its similarity to the cluster centroids. Once an assignment is made for the data, the cluster centroids are recomputed, and the algorithm persists until either a maximum number of iterations is met or until the centroids and/or memberships no longer change [68]. The pseudocode is shown as Algorithm 3.2.

Algorithm 3.2 k-means [68]

Choose a similarity or distance metric $s(\boldsymbol{x}, \boldsymbol{y})$.
Set the number of clusters k and compute their centroids $\boldsymbol{\mu}_j$, $j = 1, ..., k$.
Choose a maximum number of iterations M.
while Iterations< M **do**
    for i=1:N **do**
        Assign exemplar i to cluster $\hat{j}$ by finding the centroid with closest similarity $\hat{s}$,
        $\hat{j} = \mathbf{argmin}_{j=1,...,k} s\left(\boldsymbol{x}_i, \boldsymbol{\mu}_j\right)$.
    end for
    Recompute μ_j for $j = 1, ..., k$. If the centroids and/or memberships do not change, exit the while loop.
end while

k-medoids is a related algorithm where cluster centroids are exemplars from the dataset rather than the group means [171]. Advances in the k-means algorithm have been made both for efficiency and robustness to starting centroids. Euclidean distance is commonly used as the similarity metric, as was in this research due to investigations shown in Section 4.1. Kuang [130] greatly accelerated the Matlab® k-means function simply by noting that the distance expansion $\|\boldsymbol{x} - \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 - 2\boldsymbol{x}\boldsymbol{y}^T + \|\boldsymbol{y}\|^2$ is much faster to compute. Additionally, he removed some unnecessary computation. Elkan [69] made k-means more efficient by applying the triangle inequality to avoid unneeded distance computations.

As k-means can be sensitive to starting centroids, Bradley and Fayyad [37] developed a refinement approach based on sub-sampling. Choosing J sub-samples of the data, each sub-sample j is clustered to yield a set of centroids C_j. Next, $\bigcup_j C_j$ is clustered using each set of centroids C_j as starting centroid solutions. This yields J centroid estimates, $\overline{C}_j$, and is done to avoid solutions corrupted by outliers. The $\overline{C}_j$ with minimal squared error or distortion is chosen as best, and finally, k-means is performed for the entire dataset using this best set of centroids as its initial guess for the centroids. A comparison of normal k-means and this refined method is shown in Figure 3.8 for the Hepta dataset. As can be seen for a random centroid start with $k = 7$, the refined method correctly ascertains the classes while normal k-means does not. This was a representative result. Others have devised



(a) Normal, k=7. (b) Refined, k=7.

Figure 3.8: Refined K-Means Comparison: Hepta.

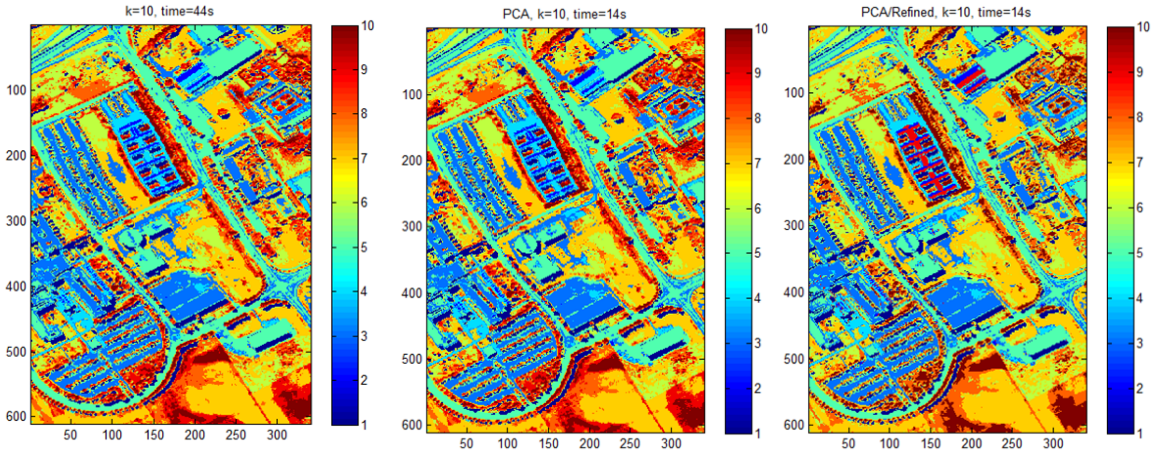
refinement starts, such as the affinity propagation start suggested by Zhu, Yu, and Jia [232].

Ding and He [65] proved that principal components are the continuous solution to the cluster membership indicators in k-means clustering. That is, the subspace spanned by the cluster centroids are given by the $k-1$ leading principal components and that this subspace

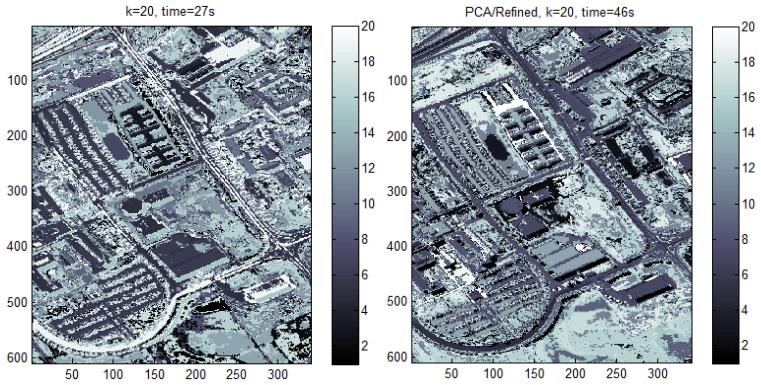
is the most discriminative. They proved this to be true for kernel k-means (k-means in the kernel space) and KPCA as well. They also proved that in the cluster subspace, between cluster distances remain nearly the same as in the original space, and that within-cluster distances are reduced. What their work implies is that exemplars can be clustered on the PCs or Kernel PCs with approximately the same result as the originating space or kernel space. This can be a powerful fact to make clustering more efficient. Fern and Brodley [72] developed an agglomerative clustering algorithm that combines the results of clusters on random projections. However, the dimensionality benefit within the clustering for this method can be achieved more directly via PCA, assuming the L_2 similarity metric. Thus, the k-means algorithm used in this research projects the data onto the first $k-1$ principal components if $k-1 < p$, utilizes Kuang's [130] code improvements, and when noted, also utilizes Bradley and Fayyad's [37] robust centroid start. Elkan's [69] concept is not used, as its implementation required checks that did not necessarily speed code execution.

Figure 3.9 depicts a comparison of these variants without and with PCA (where the $k-1$ leading components are retained) and refinement incorporated for different values of k on the Pavia University image. $k = 10$ was chosen specifically as the ground truth data contains 10 classes. As can be seen by the color-coded clusters, clustering on the components does in fact yield the same cluster membership. The value of refinement is not necessarily as obvious, but small changes can be seen where the refined clusters have broader membership for larger background classes, and more refined membership for the others. This is more obvious in Figure 3.10 for ARES1D. For $k = 20$, the refined clusters are far less noisy. This implies that the refinement could be great use when constructing skeletons, and is investigated in more detail in Section 7.3. This is in part also due to an iteration limit of $M = 100$ being used, meaning the refinement can be of benefit to improve the solution when trying to reduce run-time for very large data sets. These notably efficient

run-times are included in the sub-figure titles. In all comparisons, the same vectors were used for the initial centroid estimates.



(a) Normal, k=10. (b) PCA, k=10. (c) PCA Refined, k=10.

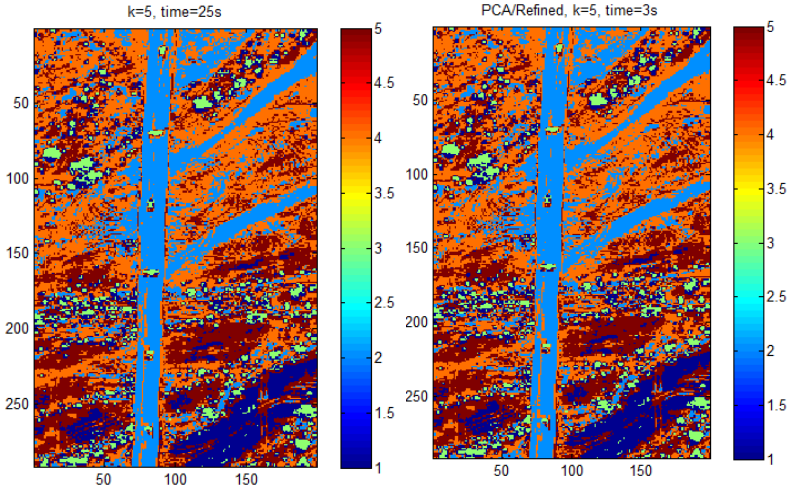


(d) PCA, k=20. (e) PCA Refined, k=20.

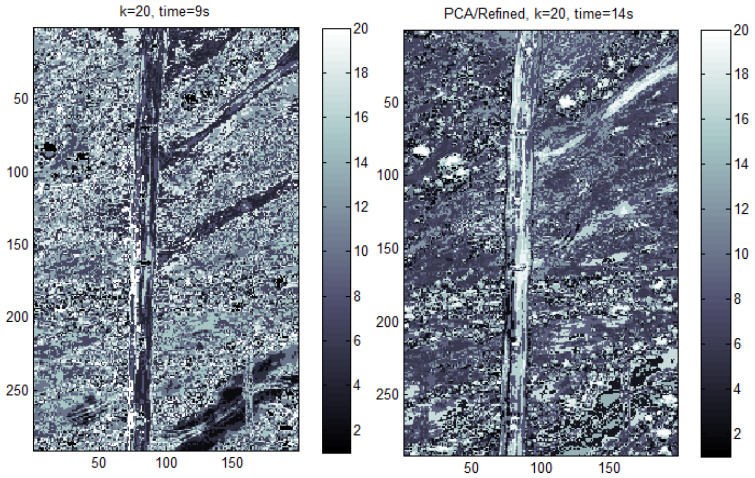
Figure 3.9: Clustering Applied to Pavia University.

3.9.2 X-means.

Due to k-means scaling poorly computationally, and because it is dependent on the choice for k, and is prone to local minima, Pelleg and Moore [172] developed the X-means



(a) PCA, k=5. (b) PCA Refined, k=5.



(c) PCA, k=20. (d) PCA Refined, k=20.

Figure 3.10: Clustering Applied to ARES1D.

algorithm. The general idea of their algorithm is to run k-means for some initial k, and then to split clusters where doing so would yield better cluster results. Specifically, for a given k-cluster model M_k the Bayes Information Criterion (BIC) is computed as,

$$BIC(M_k) = \hat{l}_k(D) - \frac{q_k}{2} \log R, \tag{3.38}$$

where $\hat{l}_k(D)$ is the log-likelihood of the set of exemplars D according to the model M_k, $q_k = k + pk$ is the number of parameters in M_k, and R is the number of exemplars in D [172]. Under an identical spherical Gaussian assumption for the clusters, the maximum log-likelihood for a specific cluster j is,

$$\hat{l}(D_j) = -\frac{R_j}{2}\log(2\pi) - \frac{R_j p}{2}\log\left(\frac{1}{p(R_j-1)}\sum_i \|x_i - \mu_{(i)}\|^2\right) - \frac{p(R_j-1)}{2} + R_j\log(R_j) - R_j\log(R). \tag{3.39}$$

Fortunately, to compute the log-likelihood for a model with multiple clusters, the cluster log-likelihoods can just be added.

The X-means algorithms begins with some number of clusters k and considers a split for each cluster in the model that has yet to evaluated. If splitting improves the BIC, then those clusters are split into two new clusters, at which point the process is repeated. Typically there is some maximum k at which to stop splitting, although ideally k converges before reaching that limit. BIC-means is a related algorithm to X-means that focuses on the BIC improvement locally by only considering the split of one cluster at a time [99]. In the version of X-means developed for this research, after each split or set of splits, k-means with the new k is run to start the next iteration. This is done until splitting no longer yields an improvement in BIC. Admittedly, this is not guaranteed to maintain parent-children cluster memberships. However, BIC-means is used for that purpose and ensures children belong to the parent cluster. It should be noted that X-means and BIC-means are still vulnerable to the limitations of k-means, in that each cluster is assumed Gaussian. Sugar and James [198] alternatively proposed locating the largest jump in distortion to detect k, but this still requires a scaling and testing of several candidates.

3.9.3 Affinity Propagation.

Frey and Dueck [73] developed the Affinity Propagation (AP) algorithm to find representative exemplars in a dataset. Input to the algorithm is a similarity matrix where

$s(i, j)$ denotes the similarity between exemplars i and j. Unlike many algorithms, $s(i, i)$ is also given an initial non-zero value to represent the preference of selecting exemplar i. The basis of AP is to pass two types of messages between exemplars. The *responsibility* $r(i, j)$ sent from i to candidate representative exemplar j reflects how well-suited j is to serve as the representative for i. The *availability* $a(i, j)$ sent from candidate j to exemplar i reflects how appropriate it is for i to pick j as its representative. These messages are updated according to the rules,

$$r(i, j) \leftarrow s(i, j) - \max_{j', j' \neq j} \{a(i, j') + s(i, j')\} \tag{3.40}$$
$$a(i, j) \leftarrow \min\{0, r(j, j) + \sum_{i', i' \notin i, j} \max\{0, r(i', j)\}\}.$$

These updates allow candidates to compete for ownership of an exemplar and for exemplars to inform what candidates are truly representative.

The self-responsibility $r(i, i)$ is set to $s(i, i)$ minus the largest of the similarities between point i and all other candidates. The self-availability is updated as,

$$a(i, i) \leftarrow \sum_{j', j' \neq i} \max\{0, r(j', i)\}. \tag{3.41}$$

After any number of iterations of these updates, $ar = a(i, j) + r(i, j)$ can be used to identify representative exemplars, where if the value of j that maximizes ar is exemplar i then i is a representative exemplar. If this is not i, then the value for j that does maximize ar identifies an exemplar that is representative for point i.

As numerical oscillations can occur in the computations during updates, the responsibilities and availabilities are damped by setting them to a factor $0 < \lambda < 1$ of the previous value plus a factor of $1 - \lambda$ of the new value, where $\lambda = 0.5$ was suggested [73]. A benefit of AP is that the number of representative exemplars does not need to be chosen *a priori*.

3.9.4 Spectral Clustering.

Spectral clustering is a two-step method, where first a Gram matrix K is formed on the data using a similarity, dissimilarity, or another kernel function. This is most often a Radial Basis Function, or exponential function [167]. Where D is the diagonal matrix of the row sums of K, K is normalized using $L = D^{-1/2}KD^{-1/2}$. Next, the projections Y of the exemplars onto the major eigenvectors of L are clustered using a traditional clustering algorithm. Ng, Jordan, and Weiss [167] also suggested re-normalizing the rows of Y to have unit length before clustering. They noted that for spectral clustering to succeed, some assumption of cluster tightness had to hold. Bengio, Vincent, and Paiement [26] showed a direct equivalence, which was alluded to in Section 3.9.1, between the KPCA mapping and spectral clustering embedding. Thus kernel k-means can also be thought of as a spectral clustering.

Many other clustering methods exist in the literature, but those discussed here are the most prevalent. Further, many of the others that had desirable properties proved to be problematic on large-dimensional data. A general category of remaining methods is hierarchal clustering. This is a general method where small clusters are merged, or large clusters are divided, repeatedly [116]. X-means can be thought of as a specific example of such an approach.

3.10 Independent Component Analysis

There are many forms of mixing models, where pixels may be treated as some mixing of a set of source materials. Independent Component Analysis (ICA) seeks directions in the feature space that are the most independent from one another [7]. Within the context of signals, assuming some set of d independent source signals $\mathbf{s}$ and sensed signals $\mathbf{x}$, the multivariate density is $p(\mathbf{s}) = \prod_{i=1}^{d} p(s_i)$. The sensed signals are some mixture of the source signals, $\mathbf{x} = A\mathbf{s}$. Here, p is temporarily used to denote the density rather than a number of features.

The independent components (ICs) are the random variables making up $\mathbf{s}$, and for the model to hold these are statistically independent and have non-Gaussian distributions. The latter assumption enables non-zero higher order statistics, which can be essential for estimation of the model. One solution strategy is to find the projection of the data that maximizes non-Gaussianity, $\mathbf{s} = W\mathbf{x}$. A is then W^{-1}. In order to simplify the process of solving for the components, Hyvärinen [104] developed a fixed-point algorithm. Assuming whitened data, FastICA uses an approximation to negentropy in order to maximize non-Gaussianity (or equivalently, minimize mutual information) [103]. Specifically, the following approximation is used:

$$J(x) = k\left(E[G(x)] - E[G(v)]\right)^2, \tag{3.42}$$

where G is some nonquadratic function that does not grow too fast, v is a Gaussian variable of zero mean and unit variance, and $k > 0$. Johnson found $G(x) = \frac{1}{4}x^4$ to work very well within the FastICA algorithm [110].

Given some number of components to estimate and initial values for W, FastICA repeats the following two steps until convergence:

1. Let $w_i^+ = E[x \cdot g(w_i^T x)] - E[g'(w_i^T x)]w_i$, where g is the derivative of G, and g' is the derivative of g.

2. Complete a symmetric orthogonalization of W.

The resulting components then define the projection. Varying forms of ICA exist, to include a Bayesian method using multi-layer perceptrons by Lappalainen and Honkela [135]. Various forms of Kernel ICA (KICA) have also been implemented using the kernel trick. Shen, Jegelka, and Gretton [188] optimized the Hilbert-Schmidt interdependence criterion using a Newton-like algorithm to yield a fast version. Bach and Jordan [18] presented algorithms based on using canonical correlation in the reproducing kernel Hilbert space. Chunhui, Yulei, and Feng [58] used KICA for anomaly detection in HSI. Both ICA and

KICA can be problematic in practice in that estimation of the independent components has a probabilistic element based on the initial estimation and the optimization convergence.

3.11 Anomaly Detection in Hyperspectral Imagery

Anomaly detection for HSI is a diverse area, where a wide array of methods have been applied. Some of the more popular detection methods that have been developed model pixels as combinations of the pixel's signature and background noise. Methods have modeled this background both locally and globally, using windows, linear methods to reduce spatial correlation, and by removing strong signatures as they are found (referred to as causal or iterative). To classify outliers from the results, combinations of histogram methods, signal-to-noise ratios (SNR), likelihood ratio tests, distances, and separation transforms have been used. Figure 3.11 depicts the general process inherent to many of these methods. Others have used mixing models as a precursor to consider pixels as some combination of a set of *endmembers* in order to remove redundant information. In most cases, spectral detection is the primary focus as an anomaly may occupy only part of a pixel, and detecting solely on spatial information requires a target to be large relative to pixel size. Robustness towards false positives or soft anomalies can be an important, yet difficult task.

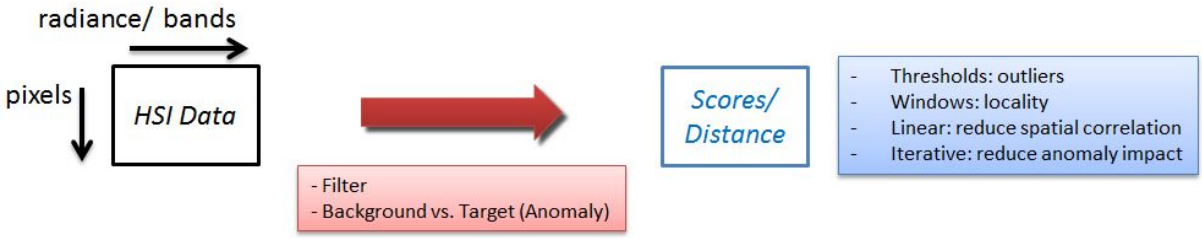


Figure 3.11: RX-Like Detectors.

Many existing anomaly detectors are parametric or at least make some distributional assumptions. Although these are typically simple and unsupervised, they can have a large disadvantage in that they often produce an intolerable high number of detections per scene [180]. For example, the Reed-Xiaoli (RX) algorithm assumes background pixels in a local neighborhood around the pixel under test are independent identically distributed Gaussian random variables, but it has been shown empirically that multiple classes of terrain in the local background do not satisfy this assumption [22]. Further, Frontera-Pons et al. [75] stated that in HSI the actual response of a detector to background pixels differs from the theoretically predicted distribution for Gaussian backgrounds, and that the empirical distribution typically has heavier tails that influence the observed false-alarm rate of a detector. Non-parametric methods have been developed as an alternative, but are typically based on some overarching hypothesis test. Wavelets have been applied so as to reduce correlation and in order to use both spatial and spectral information.

3.11.1 RX-Based and Uniform Detectors.

One popular class of algorithms for finding anomalies in HSI is based on the RX anomaly detector developed by Reed and Yu [177]. The standard RX algorithm moves a window through the image in order to compute local background estimates to compare with center pixel under test. This is popular due to its simplicity, but there can be a trade-off with its computational expense depending on how exactly the pixel or local window under test is compared to the current background estimate. Other methods, sometimes referred to as uniform detectors or filters, also make a simple determination based on comparing a pixel or local window under test against a current background estimate. These methods, unlike the quadratic detector and matched filters, do not assume that information is known about the anomaly class [154].

3.11.1.1 RX-Based Detectors.

The basic RX algorithm models the image background as a Gaussian distribution with zero mean and an unknown covariance matrix which can be estimated globally or locally from the data. Any anomaly is modeled as a linear combination of the anomaly signature and the background noise. Therefore, an anomaly spectral signature is represented by a Gaussian distribution with a mean equal to that of the signature of the anomaly and an additive noise equal to the background covariance matrix [195]. The detection process is then based on exploiting the difference between the spectral signatures of a pixel $\boldsymbol{x}$ and its surrounding pixels, in actuality the squared Mahalanobis distance,

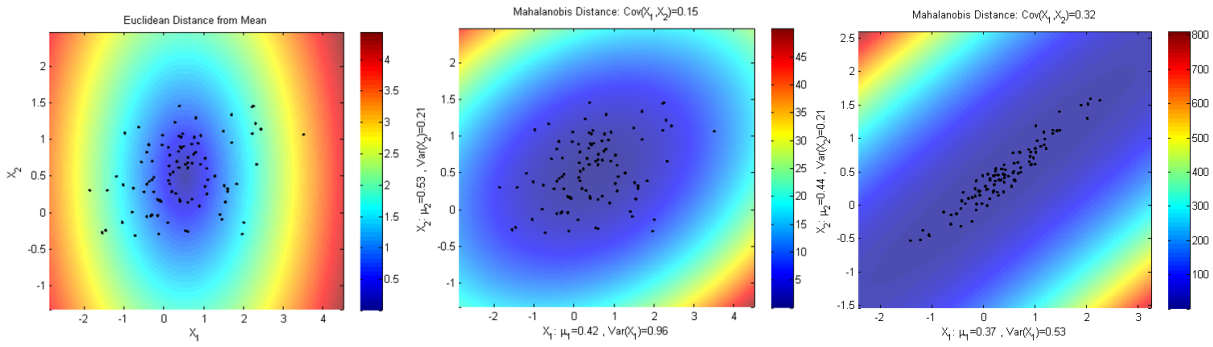
$$\delta_{RX}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{\mu})^T C_{p\times p}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}), \tag{3.43}$$

where $\boldsymbol{\mu}$ is the global mean spectral vector (over the bands) and C is the sample spectral covariance matrix. In the sense that small eigenvalues of C correspond to a large value for $\delta_{RX}(\boldsymbol{x})$, Soobaf, et al. [195] noted this is an inverse operation to Principal Component Analysis, where the search is for anomalies in minor components. This is interesting because it assumes that pixels that occur with low probability in the data do not show in major principal components. Stated differently, variances of anomalies contribute to the sample variance in a minor fashion, and thus Hsueh and Chang [100] noted that standard use of Principal Components could be ineffective in finding anomalies. Using spectral decomposition of the covariance matrix, C can be decorrelated into a diagonal matrix Λ such that $V^T C_{p\times p} V = \Lambda$, where V is a unitary matrix. Then,

$$\boldsymbol{x}^T C_{p\times p}^{-1} \boldsymbol{x} = \sum_{l=1}^{p} \lambda_l^{-1} y_l^2, \tag{3.44}$$

where $\boldsymbol{y} = V^T \boldsymbol{x}$ [46]. We can see that the RX detector can be interpreted as a matched filter, $\kappa \boldsymbol{d}^T \boldsymbol{x}$ operating on $\boldsymbol{x} - \boldsymbol{\mu}$ with the matched signal $\boldsymbol{d} = (\boldsymbol{x} - \boldsymbol{\mu})^T C_{p\times p}^{-1}$ and the scale constant $\kappa = 1$. Thomas [204] showed that the RX detection test came naturally from the Maximum Likelihood when considering the spectral and spatial domains as continuous.

The use of Mahalanobis distance can be powerful in that it considers the variance structure of the data. Figures 3.12(a) and 3.12(b) compare Euclidean and Mahalanobis distance for a common set of data, while Figure 3.12(c) shows the difference in distance shape when the variance structure is changed.



(a) Euclidean - Data 1. (b) Mahalanobis - Data 1. (c) Mahalanobis - Data 2.

Figure 3.12: Distance Comparison on Two Data Sets.

Given PCA, Lee and Landgrebe [137] noted for multispectral and hyperspectral data that the first few eigenvectors often contain much of the variance, and under a Gaussian assumption the data is then an elongated hyperellipsoid. Under these conditions, they argued for the use of second-order (and higher) statistics for a classifier, as is done with RX.

Two simple variants of the basic RX algorithm are the normalized and modified RX methods. These divide $\delta_{RX}(\boldsymbol{x})$ by $\|\boldsymbol{x}-\boldsymbol{\mu}\|$ and $\|\boldsymbol{x}-\boldsymbol{\mu}\|^2$, in essence changing κ [46]. When the percentage of anomalous pixels is relatively large, the sample covariance matrix no longer represents the background distribution. In this case, the weighted RX algorithm (WRX) developed by Ren, Chen, and Chen [178] assigns a weight to each pixel in the sample

covariance matrix using its distance to the data center. The weighted covariance matrix is,

$$C_w = \frac{\sum_{i=1}^{N} w_i(\boldsymbol{x}_i - \boldsymbol{\mu}_w)(\boldsymbol{x}_i - \boldsymbol{\mu}_w)^T}{\sum_{i=1}^{N} w_i}, \tag{3.45}$$

where $\boldsymbol{\mu}_w = \frac{\sum_{i=1}^{N} q_i \boldsymbol{x}_i}{\sum_{i=1}^{N} q_i}$, $q_i = \frac{1}{1 + \|\boldsymbol{x}_i - \boldsymbol{\mu}\|}$, and $w_i = \frac{1}{1 + \|\boldsymbol{x}_i - \boldsymbol{\mu}_w\|}$ [195]. The RX filter is then,

$$\delta_{WRX}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{\mu}_w)^T C_w^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_w). \tag{3.46}$$

Ashton and Schaum [17] showed that background subtraction could enhance the RX detection algorithm. This led to the following detector [46],

$$\delta_{RXD-UTD}(\boldsymbol{x}) = (\boldsymbol{x} - \mathbf{1})^T C_{p\times p}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}). \tag{3.47}$$

If the spectral properties of an anomaly can be characterized only by first-order statistics, then a better choice is to use the sample correlation matrix $R_{p\times p}$ instead of $C_{p\times p}$ and $\boldsymbol{x}$ instead of $\boldsymbol{x} - \boldsymbol{\mu}$ in the various detectors to allow for first-order and second-order statistics [46]. To enable "real-time" computation, computation of $R_{p\times p}^{-1}$ is done using QR-decomposition, processing a pixel as it is received. This is called Causal RX (CRX),

$$\delta_{CRX}(\boldsymbol{x}_k) = \boldsymbol{x}_k^T (R_{p\times p}^{-1}(\boldsymbol{x}_k)) \boldsymbol{x}_k, \tag{3.48}$$

where $\boldsymbol{x}_k$ denotes processing up to the k-th pixel. This is also better than using the covariance matrix, as C^{-1} required computation of the mean for the entire image. Here, the information used for data processing is up to the pixel being processed and updated on pixels already processed [195]. If a detected anomaly with a strong signature remains, it can dominate and obscure anomalies that could not be detected subsequently. To address this issue, Hsueh and Chang [100] devised the Adaptive Causal Anomaly Detection algorithm (ACAD). This algorithm is the same as CRX, except the causal sample correlation matrix removes all detected anomalies up to the current pixel vector $\boldsymbol{x}_k$.

Given the inherent assumption of equal covariance matrices for the two hypotheses in the RX algorithm, for sub-pixel targets it may be better to assume that the background

has the same covariance structure but different variance. This approach led to the Adaptive Coherence/Cosine Estimator (ACE) detector,

$$D_{ACE}(\boldsymbol{x}) = \frac{\boldsymbol{x}^T C_{p\times p}^{-1} S \left(S^T C_{p\times p}^{-1} S\right)^{-1} S^T C_{p\times p}^{-1} \boldsymbol{x}}{\boldsymbol{x}^T C_{p\times p}^{-1} \boldsymbol{x}}, \tag{3.49}$$

where S is a known representation of the target subspace [154]. The detector and hypotheses can be adjusted accordingly if the background is modeled by a subspace model, assuming the subspace models are known. Wang et al. [213] used the relative fluctuations in correlation coefficients for adjacent spectral bands to detect the modes in HSI pixel signatures. Each of these was used to yield a band subspace, or a subset of bands, on which to perform PCA. After projection, they used RX to detect anomalies for each subset and fused the results for a final prediction.

In RX algorithms, in general, it is assumed that the background and target have the same covariance matrix. This is not necessarily valid when trying to detect a particular target, but is difficult to avoid as the statistical structure of any anomaly is undefined. To take advantage of the spectral correlation provided in the RXD, Chang [45] suggested using distances other than Mahalanobis, such as Bhattacharyya distance, to distinguish between anomalies. Further variants of the RX algorithm exist, and a few of these and their aspects are covered later in this chapter.

3.11.1.2 Low-Probability Detection Method.

Another basis method is the Low-Probability Detection method (LPD) [94]. Unlike the RX detector (where the current pixel is used as the matched signal), LPD uses the unity vector as the matched signal. If the sample correlation matrix is replaced with the sample covariance matrix, a uniform target detector (UTD) can be defined as,

$$\delta_{UTD}(\boldsymbol{x}) = (\mathbf{1} - \boldsymbol{\mu})^T C_{p\times p}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}). \tag{3.50}$$

Here, no information is introduced to the detector, but an anomalous target is assumed to have radiance uniformly distributed over all of the spectral bands, thus uniformly distributed background signatures are extracted.

3.11.1.3 Kernel RX.

Kwon and Nasrabadi [133] developed the Kernel RX (KRX) algorithm utilizing KPCA inside of the RX algorithm. They noted that in general, the Gaussian assumption in the RX-algorithm is not valid. Therefore, they formulated a non-linear version where a Gaussian distribution for the two hypotheses was assumed in the higher-dimensional feature space. Thus, modeling the input data in this new feature space by a Gaussian distribution was equivalent to representing the distribution of the input data with a more complex model in the original space.

This yielded the RX-algorithm in the feature space,

$$RX\left(\Phi(\boldsymbol{x})\right) = \left(\Phi(\boldsymbol{x}) - \mu_{b_\Phi}\right)^T C_{b_\Phi}^{-1} \left(\Phi(\boldsymbol{x}) - \mu_{b_\Phi}\right), \tag{3.51}$$

where C_{b_Φ} is the estimated covariance and μ_{b_Φ} is the estimated mean of the background clutter samples in the feature space.

As with KPCA, this RX-algorithm can be implemented without explicitly using Φ. Consider the eigen-decomposition $C_{b_\Phi} = V\Lambda V^T$ where V is the set of eigenvectors and Λ is the diagonal matrix of eigenvalues for C_{b_Φ}. Then, $C_{b_\Phi}^{-1} = V\Lambda^{-1}V^T$. Thus, the KRX equation becomes,

$$RX\left(\Phi(\boldsymbol{x})\right) = \left(\Phi(\boldsymbol{x}) - \mu_{b_\Phi}\right)^T V\Lambda^{-1}V^T \left(\Phi(\boldsymbol{x}) - \mu_{b_\Phi}\right). \tag{3.52}$$

$\left(\Phi(\boldsymbol{x}) - \mu_{b_\Phi}\right)^T V$ and $V^T\left(\Phi(\boldsymbol{x}) - \mu_{b_\Phi}\right)$ are just the kernel component scores. Recall from Equation 3.10, these can be computed using the normalized eigenvectors of the centered Gram matrix formed on the training data and the kernel values. Thus, only Λ^{-1} is needed to complete the calculation. In fact, it is easy to show that $\Lambda^{-1} = N\Omega^{-1}$ where Ω^{-1} is the diagonal matrix of non-zero eigenvalues of the centered Gram matrix.

The General Likelihood Ratio Test can still be applied, and the kernel function used needs to be valid so as to yield a positive semi-definite Gram matrix [133]. To try and better meet the Gaussian assumption in the feature space, Kwon and Nasrabadi [133] used a Gaussian kernel and they noted that the eigenvalues decayed quickly and that the kernel is translation-invariant. They also estimated the kernel matrix $\hat{K}_b$ globally and locally for comparison. To estimate globally, k-means clustering was used to find centroids from which to estimate the matrix. They used $k = 600$ so as to fully represent the background and on their image of interest, found $\sigma = 40$ to work well. To estimate the kernel matrix locally, a dual cocentric window approach was used. In both cases, they scaled the data by the largest spectral value to try and best utilize the Gaussian kernel. They did not provide an approach on selecting a subset of kernel principal components to use for the KRX scores, but showed favorable detection improvement over linear-based RX algorithms [133, 165]. Nasrabadi [165] presented KRX as an example of the more general linear subspace-based anomaly detector. For some projection matrix W and outer window, the associated projection separation statistic is,

$$s = (\boldsymbol{x} - \mu_{out})^T \, WW^T \, (\boldsymbol{x} - \mu_{out}) \, . \tag{3.53}$$

3.11.1.4 General Likelihood Ratio Test.

One remaining aspect of the RX detector is the actual identification of an anomaly. In its simplest form, a pixel can be determined as an anomaly if the RX score is greater than $\chi^2_{\alpha,p}$ where α and p are the corresponding quantile and degrees of freedom of the Chi-squared distribution.

To develop this further, consider the formulation of the hypothesis test. Assuming $H_0 : \theta = \theta_0$ and $H_a : \theta = \theta_a$ based on a random sample from a distribution with parameter θ, let $L(\theta)$ be the likelihood of the sample when the value of the parameter is θ. The Neyman-Pearson Lemma states that for a given α, the test that maximizes the power at θ_a has a rejection region determined by $\dfrac{L(\theta_0)}{L(\theta_a)} < t$. t is chosen so that

the test has the desired value for α. This is a most powerful α-level test for H_0 versus H_a [209]. Relating this to the hyperspectral problem, let the probability of detection be $P_D = P(H_a; H_a) =$ probability decide H_a when H_a is true, and let the probability of false alarm be $P_{FA} = P(H_a; H_0) =$ probability decide H_a when H_0 is true. Here, H_0 is relative to the background and H_a is relative to an anomaly. Using Neyman-Pearson, P_D is maximized subject to a desired fixed $P_{FA} = \alpha$, where the likelihood ratio test decides H_a if the likelihood ratio [64],

$$L(x) := \frac{p(\mathbf{x}; H_a)}{p(\mathbf{x}; H_0)} > \gamma. \tag{3.54}$$

γ is found from,

$$P_{FA} = \int_{\{\mathbf{x}: L(\mathbf{x}) > \gamma\}} p(\mathbf{x}; H_0)\, d\mathbf{x} = \alpha. \tag{3.55}$$

This approach is referred to as the Generalized Likelihood Ratio Test (GLRT).

A Bayesian approach could minimize the Bayesian risk (cost function) for arbitrary costs C for deciding one hypothesis when the other is true, but this requires prior probabilites for each hypothesis. Bayes Risk is,

$$\mathcal{R} = E[C] = \sum_{i=0}^{1} \sum_{j=0}^{1} C_{ij} P\left(H_i | H_j\right) P\left(H_j\right), \tag{3.56}$$

where $i, j = 0$ reflects the null hypothesis and $i, j = 1$ reflects the alternative hypothesis. In RX detectors, these are background and signal plus background, respectively. Assuming $C_{10} > C_{00}$ and $C_{01} > C_{11}$, the detector that minimizes $\mathcal{R}$ decides H_a if $\frac{p(\mathbf{x}|H_a)}{p(\mathbf{x}|H_0)} > \frac{(C_{10} - C_{00}) P(H_0)}{(C_{01} - C_{11}) P(H_a)} = \gamma$ [64]. This Bayesian framework encompasses both minimum probability of error or maximum a posteriori, P_E or MAP, and maximum likelihood (ML). For the former, $C_{ii} = 0$, $C_{ij} = 1$ for $i \neq j$. H_a is decided if:

$$\begin{aligned} \min P_E: &\quad \frac{p(\mathbf{x}|H_a)}{p(\mathbf{x}|H_0)} > \frac{H_0}{H_a} = \gamma, \\ \text{MAP}: &\quad P(H_a|\mathbf{x}) > P(H_0|\mathbf{x}). \end{aligned} \tag{3.57}$$

For ML, the costs are the same, but all priors $P(H_i)$ are equal. The detector decides H_a if $P(\mathbf{x}|H_a) > P(\mathbf{x}|H_0)$. Both Neyman-Pearson and this Bayes approach can be expanded to

more than two hypotheses. This latter Bayesian approach does not appear to be prevalent in the literature.

Thomas [204] used functional statistics to model anomaly detection using a continuous spectral domain approach. He imposed Gaussian behavior and developed a hypothesis test where the null hypothesis was the presence of only background noise, and the alternative was the presence of a spatially patterned signal with added noise. Utilizing a likelihood ratio under his model, he found optimality to be analogous to the GLRT as used in RX.

3.11.1.5 Windows.

The RX-based algorithms are known as local or global depending on if the mean spectrum is derived from all of the image data or from a local window around each pixel during anomaly detection. Other methods also make use of windows to try and separate anomalies from background. As scenes often have multiple land covers, the underlying distributions are usually multimodal. Therefore, windows may sub-divide the scene such that each class can be characterized by a unimodal distribution. Kwon, Der, and Nasrabadi [131] developed the Dual Window-base Eigen-Separation Transform detector (DWEST) using two local windows. These inner and outer windows are designed to maximize the separation between anomalies and the background in a low-dimensional subspace, albeit the exact formation of these windows is subjective. Typically, an inner window is meant to be large enough to encapsulate a target. As these local windows are moved through the image, a local mean (μ_{in}, μ_{out}) and covariance matrix (C_{in}, C_{out}) of each window are formed and their differences taken. Figure 3.13 shows the general idea of a dual window approach.

In DWEST, anomalies are extracted by projecting the differential mean onto the eigenvector Λ_i or set of eigenvectors with the largest positive eigenvalues $\vec{\lambda}$ of the differential covariance

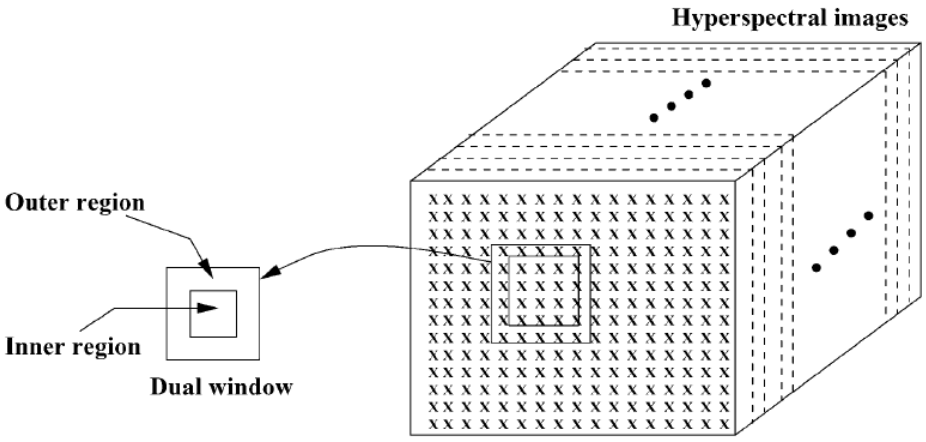


Figure 3.13: Dual Cocentric Window [133].

matrix $C_{diff} = C_{in} - C_{out}$:

$$\delta_{DWEST} = |\sum_{i} \Lambda_i^T \left(\mu_{out} - \mu_{in}\right)(\boldsymbol{x})|. \tag{3.58}$$

RX can also be implemented using the two-window approach simply by using:

$$\delta_{RX-DW}(\boldsymbol{x}) = |(\mu_{out} - \mu_{in})(\boldsymbol{x})^T \left[C_{out}^{-1}(\boldsymbol{x})\right](\mu_{out} - \mu_{in})(\boldsymbol{x})|. \tag{3.59}$$

Kwon, Der, and Nasrabadi [131] argued that the eigenvectors associated with a small number of the large positive eigenvalues could successfully extract spectrally distinctive materials present in the inner window.

Liu and Chang [144] used three nested windows in the Nested Spatial Window-base Target Detector (NSWTD). An inner and middle window are used to extract the smallest and largest anomalies, respectively, and the outer window is used to model the local background. Instead of using Mahalanobis distance or an eigen-separation transform, Orthogonal Projection Divergence (OPD) is used, given between two instances as:

$$OPD\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \sqrt{\left(\boldsymbol{x}_i P_{\boldsymbol{x}_j}^{\perp} \boldsymbol{x}_i + \boldsymbol{x}_j P_{\boldsymbol{x}_i}^{\perp} \boldsymbol{x}_j\right)}, \tag{3.60}$$

where $P^{\perp}_{\boldsymbol{x}_k} = I_{p\times p} - \boldsymbol{x}_k \left(\boldsymbol{x}_k^T \boldsymbol{x}_k\right)^{-2} \boldsymbol{x}_k^T$. First, OPD is implemented between inner and middle windows, as:

$$\delta_1^{2W-NSW}(\boldsymbol{x}) = OPD\left(\mu_{in}(\boldsymbol{x}), (\mu_{out} - \mu_{in})(\boldsymbol{x})\right). \tag{3.61}$$

The second implementation is between middle and outer windows,

$$\delta_2^{2W-NSW}(\boldsymbol{x}) = OPD\left(\mu_{mid}(\boldsymbol{x}), (\mu_{out} - \mu_{mid})(\boldsymbol{x})\right). \tag{3.62}$$

The final thee-window NSWTD is,

$$\delta^{3W-NSW}(\boldsymbol{x}) = max_{i=1,2}\left[\delta_i^{2W-NSW}(\boldsymbol{x})\right]. \tag{3.63}$$

In a dual-window setting, Nasrabadi [165] noted that PCA would not exploit information in the inner and outer windows simultaneously, and so proposed using the linear discriminant or kernel linear discriminant to find an optimal direction to discriminate between inner and outer window samples.

As another method similar to the DWEST algorithm, the Eigenspace Separation Transform (EST) finds the eigenvectors and values of the difference correlation matrix, which is the difference of the correlation matrices of the inner and outer windows. W is then chosen as the set of d eigenvectors corresponding to the most positive or most negative eigenvalues. Typically, this choice is made by choosing the set of eigenvalues (positive or negative) with the largest absolute sum [205]. EST can also be adapted to a higher-dimensional feature space by using the difference correlation matrix in the feature space [78].

3.11.1.6 Iterative RX.

The iterative RX detector (IRX) by Taitano, Gaier, and Bauer [199] calculates improved estimates of the mean vector and covariance matrix of the background pixels over several iterations until the set of anomalies converges or a maximum number of iterations is reached. Each iteration the standard RX algorithm is run, yielding a score for each testable pixel in the image. Pixels selected as anomalies in the previous iteration are excluded from

the data used to estimate the mean vector and covariance matrix of the background in that iteration, and the process is repeated. This process is used to help eliminate the effect of outliers on the mean and covariance estimates used for the RX-statistic. This was done using a window.

3.11.1.7 Linear RX.

Linear RX (LRX) and Iterative LRX (ILRX) are similar to RX and IRX, respectively, but instead of a window being moved through the image, a vertical line of pixels above and below the current pixel is used [216, 217]. If the number of required pixels above or below the pixel under test exceeds the height of the image, pixels are taken from the bottom of the previous column or from the top of the following column. The use of this linear approach is to increase the average distance between those pixels being used to estimate the background and thus to mitigate spatial correlation. This also allows for reduction of bias and error in the estimation of the background mean and covariance. This concept is shown in Figure 3.14. Taitano, Geier, and Bauer [199] also explored using these methods on the Principal Components of the line of pixels.

3.11.2 Topology Anomaly Detector.

The Topology Anomaly Detector (TAD) was developed by Basener, Ientilucci, and Messinger [23]. First, a random sample of reasonable size is taken from the HSI image for the purpose of modeling the background. The distance between every pair of pixels in this sample is computed and a graph is formed by adding an edge between the closest $b_1\%$ of pairs of pixels. From these adjacencies, the connected components (sub-graphs) of the graph are found. These components are meant to represent materials in the image [66]. The largest components containing greater than $b_2\%$ of the sample pixels are designated as background, where b_2 is assumed to also be the percentage of pixels in the image that are background. In practice, $b_1 = 10$ and $b_2 = 2$ were recommended, citing network theory and properties of components for the former [24].

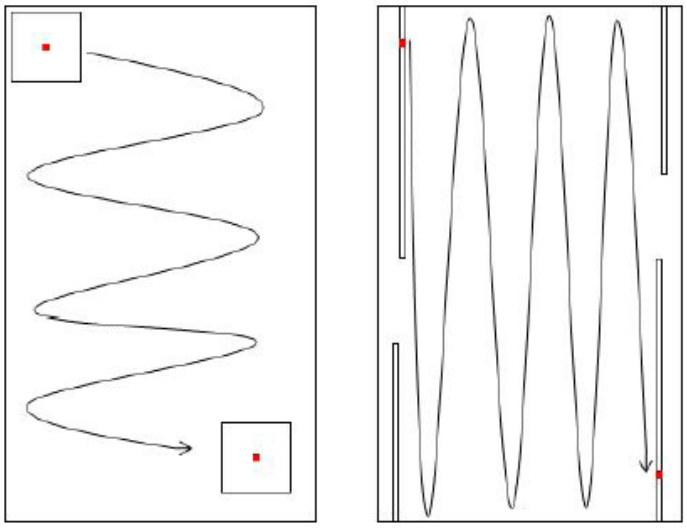


Figure 3.14: Window vs. Line.

The pixels in the components, being well-connected, are meant to form a topological model for the image background. In order to measure each pixel against the background, codensity is used. The k-th codensity δ_k for a pixel in the image is the distance to the k-th nearest neighbor. The TAD ranking is defined as,

$$TAD(x) = \sum_{i=3}^{5} \delta_i(x). \tag{3.64}$$

This ranking is used to provide level sets of arbitrary topology and to allow detection of pixels in the holes of the background convex hull [24]. Additionally, the codensity enables pixels near background regions of low density to have a higher score than those near regions of high density. In order to make a final anomaly decision for each pixel, the TAD ranking is thresholded.

3.11.3 Autonomous Global Anomaly Detector.

One algorithm of significant interest to compare to is the Autonomous Global Anomaly Detector (AutoGAD), originally developed by Johnson [110]. It is a fully

unsupervised algorithm and is a combination of several methods. A general overview of the algorithm is shown in Figure 3.15.

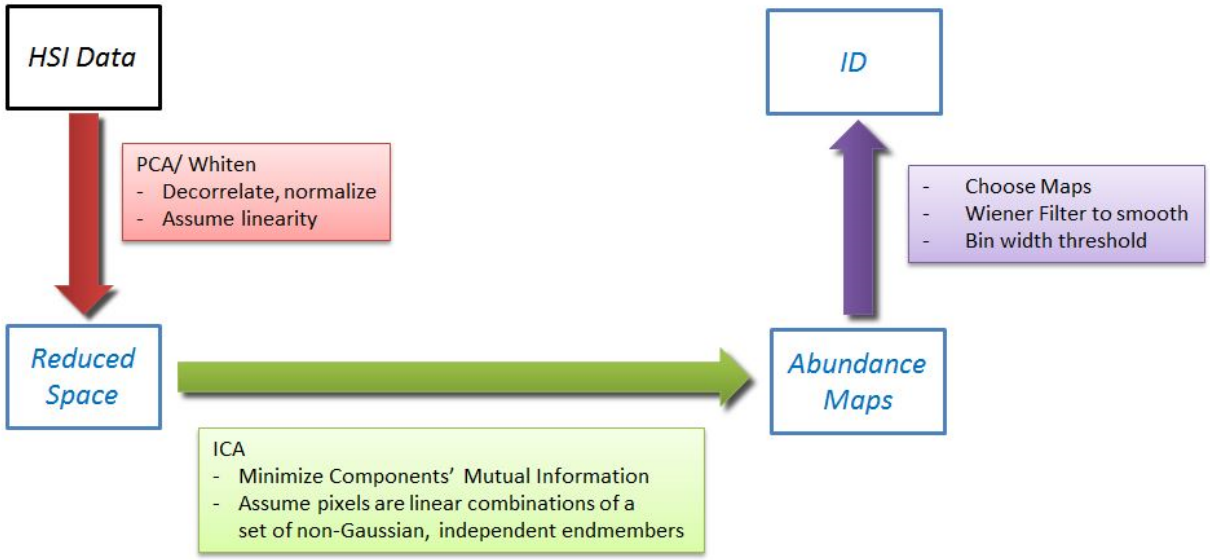


Figure 3.15: AutoGAD.

First, the data matrix is made into a pixel×band matrix, where each pixel vector is made into a row in the data matrix. This is depicted in Figure 3.16. This matrix is reduced through PCA and the components are whitened. In order to determine the number of components to retain, the knee of the eigenvalue curve relative to the maximum distance from the log secant line is found. This is referred to as Maximum Distance Secant Line (MDSL) and the concept is shown in Figure 3.17.

In order to provide better components, given the likely non-Gaussian nature of the HSI data, ICA is performed beginning with the scores of the retained principal components. In order to select the resulting maps to use to actually flag the anomalies, maximum pixel scores on the component are used to nominate components. Additionally, a threshold on

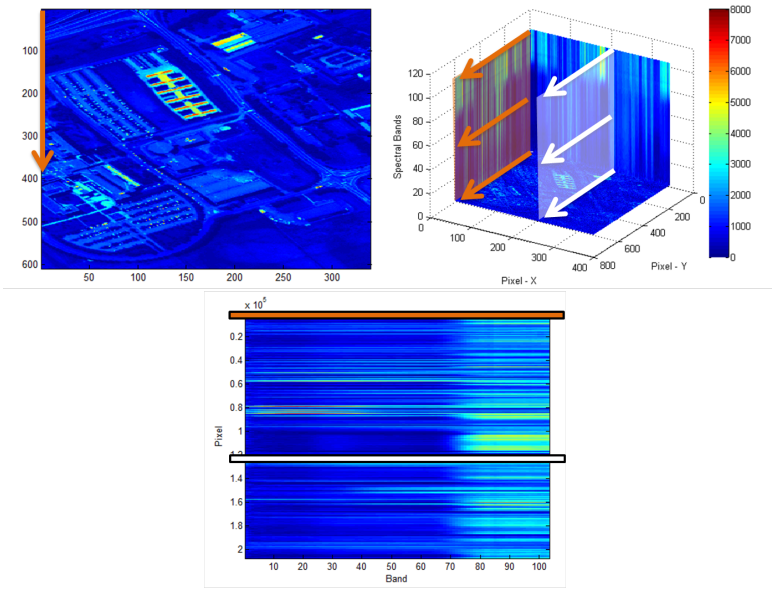


Figure 3.16: Pixel×Band Representation.

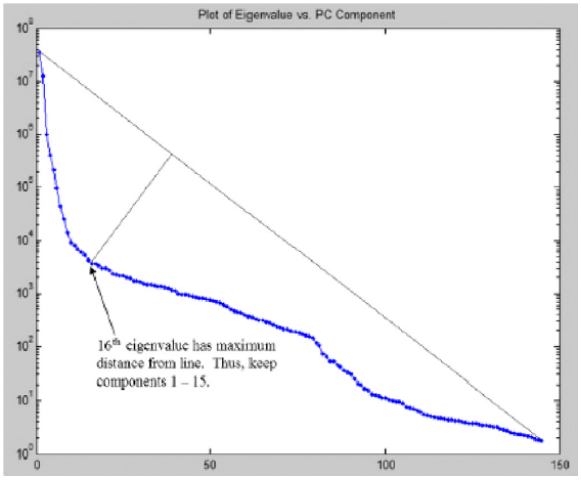


Figure 3.17: Finding the Eigenvalue Cutoff [111].

the potential anomaly signal-to-noise ratio (PA SNR),

$$\text{PA SNR} = 10 \cdot \log_{10}\left(\frac{\text{Var(potential anomaly signal)}}{\text{Var(background)}}\right), \tag{3.65}$$

is used to nominate components, using mappings where pixels exceed the threshold. For calculation of this ratio, pixels are split into background and potential anomalies through the use of a first zero-bin detection histogram, similar in nature to the method later described in Section 3.11.6.1. The potential anomalies are those that come after the first empty zero-point bin from the histogram constructed from the signal, given some bin width.

Finally, given the mappings exceeding both a score and SNR threshold, an iterative window-based Wiener filter can be applied in order to smooth the mappings, reducing background noise and making anomalies more recognizable. The Wiener filter uses a square window to make a neighborhood estimate for the mean and standard deviation, and to estimate local Gaussian noise before filtering. In order to make anomalies more recognizable, several iterations of the filter may be required. Johnson [111] chose this method as it filters more heavily where the variance is close to system noise. The histogram method is then used again for the final classification.

AutoGAD has been shown to have desirable results on several HYDICE images versus algorithms such as RX, ILRX, and Support Vector Data Description, but it also involves several user-defined parameters [111]. These include the bin-width(s) for the histogram, the PA SNR threshold, the component score threshold, an adjustment to the dimensionality assessment from MDSL, and several associated with the adaptive Wiener filter (such as window size and number of iterations). Williams [216] investigated the use of Robust Parameter Design (RPD) to better optimize these parameters for the HYDICE ARES images.

Johnson [111] used Hyvärinen's [104] FastICA to solve the ICA problem, using the contrast function $G(u) = \frac{1}{4}u^4$ with its derivative $g(u) = u^3$. However, Hyvärinen [104] noted that the kurtosis function was justified on statistical grounds only for estimating sub-Gaussian independent components when there were no outliers. Jablonski [107] later developed an algorithm inspired by AutoGAD where ICA was removed and obvious

outliers removed after a first iteration, among other changes. This is presented in Section 3.11.4.

Jablonski [107] also noted an inconsistency in true positive rates in the results presented by Johnson, Williams, and Bauer [111]. This led him to develop a method to generate Receiver Operating Characteristic curves (ROC) for AutoGAD. He proposed multiplying the zero-bin detection thresholds by a nominal factor that could vary. The results led him to find that the bin widths needed to be adjusted according to the range of the mapping scores and the number of pixels in the image. He then replaced the bin width parameters with a number of pixels per bin parameter Y. This enabled the bin width ω to vary for any given mapping as,

$$\omega = \frac{Y}{N} \left(\max(scores) - \min(scores) \right). \tag{3.66}$$

Use of the Y parameter enabled him to yield better and more consistent results [107].

3.11.4 Multiple Principal Component Analysis.

Inspired by AutoGAD, IRX, reconstruction error-based methods, and general investigation of the HYDICE ARES imagery, Jablonski [107] developed an autonomous global anomaly detector named Multiple PCA (MPCA). Using an ensemble of four scores derived from PCA, he used an iterative technique to remove potential anomalies from an initial covariance estimate. The ensemble was used to make the response more robust to different images and targets. These scores, $D_1 : D_4$, and a general overview of the algorithm are shown in Figure 3.18, where Q denotes reconstruction error as previously explained in Equation 3.1. Here, the zero-histogram detection is again the first zero-bin threshold as used in AutoGAD.

In reality, the algorithm is more complicated than Figure 3.18. Jablonski [107] found the Wiener filter and PA SNR from AutoGAD to also provide benefit in decreasing false positives. The pseudocode to match his archival code is shown as Algorithm 3.3.

Algorithm 3.3 MPCA[107]

1: Remove absorption/noisy bands and reshape the data cube to $N \times p$.

2: $X^S_{N\times p} = \left(X_{N\times p} - \mathbf{1}_{N\times 1}\boldsymbol{\mu}^T\right) D^{-1/2}$: data is centered and standardized.

3: Find eigenvectors V and eigenvalues Λ from $cov\left(X^S_{N\times p}\right)$: do PCA.

4: Use MDSL to determine the dimensionality k. Let $T_{N\times k}$ denote the PC scores for the major k components and $T_{N\times p}$ denote all PC scores.

5: Let $E = X^S_{N\times p} - T_{N\times k}V^T_{p\times k}$. $D_{3_i} = Q_i = \sum_{j=1}^{p} E^2_{ij}$: squared reconstruction error based on the first k components.

6: $\hat{T}_{N\times p} \Leftarrow Wiener\left(T_{N\times p}\right)$: use window-based Wiener filter.

7: $Z_{N\times p} = \hat{T}_{N\times p}\Lambda^{-1/2}$: standardize/whiten the scores. $\hat{Z} = Z \circ Z$: square each element.

8: $D_1 = \sum_{j=1}^{k} \hat{Z}_{*j}$: sum the major squared scores. $D_2 = \sum_{j=k+1}^{p} \hat{Z}_{*j}$: sum the minor squared scores. $D_{4_i} = median(\hat{Z}_{i*})$: find the median for each row.

9: $D_t \Leftarrow D_t/\sigma_t$ for $t = 1, 2, 3$: divide by the standard deviation of its vector.

10: $D_t \Leftarrow Wiener\,(D_t)$ for $t = 2, 3, 4$: use window-based Wiener filter.

11: $D_t \Leftarrow \sqrt{D_t}$ for $t = 1, 2, 3, 4$: take the square root of every element in each score vector.

12: Construct histograms for D_t, $t = 1, 2, 3, 4$, and find first zero-bin using pixels per bin parameter $Y_{initial}$. $D_t > \nu_{t\,\text{Location of 1st empty histogram bin}}$ are voted as anomalies (if true for any single score).

13: **if** Any voted anomalies **then** (Begins 2nd Iteration)

14: Remove anomalies from $X^S_{N\times p}$. Do Step 3, and then Step 4 using previous k.

15: **end if**

16: Do Steps 5-11.

17: Construct histograms for D_t, $t = 1, 2, 3, 4$, and find first zero-bin using pixels per bin parameter for each D_t, Y_{final_t}. $D_t > \nu_{t\,\text{Location of 1st empty histogram bin}}$ are voted as potential anomalies.

18: If a pixel has greater than one vote on $D_{1:4}$ then the pixel is classified anomalous.

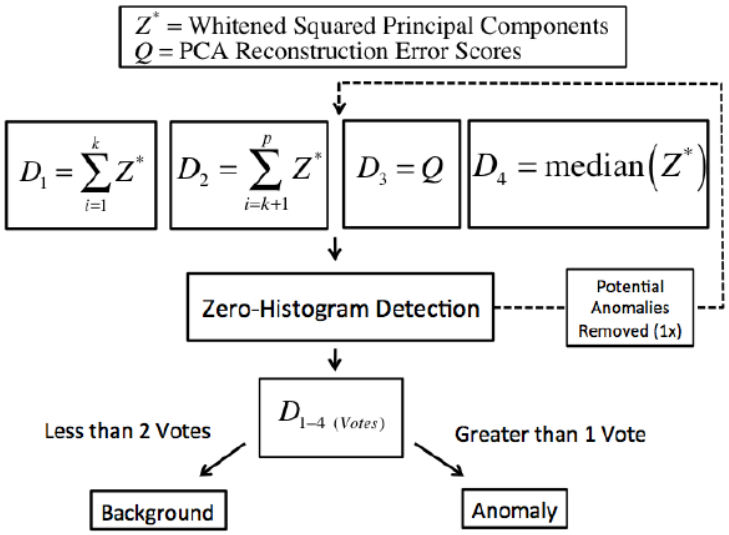


Figure 3.18: MPCA Overview [107].

Jablonski [107] chose the D_1 score to find anomalies that inflate variance and covariances, while D_2 would catch anomalies that do not or that are inconsistent with the covariance. Realizing the noise present in some of the minor components, he used D_4 to be less sensitive to this. In fact, he showed the relation of D_4 to Mahalanobis distance. D_3 was chosen to serve as an alternative to D_2 where equal weighting is not given to all of the principal component directions through whitening [107]. In order to determine settings for the parameters, to include the pixels per bin parameters, he fit a second-order model to an objective designed to maximize the average True Positive rate and minimize the average and variance of the False Positive rate.

3.11.5 BACON and Other Detector Types.

Other anomaly detection methods exist in the literature, including more where projections are used to dimension reduce and uncover latent structure. One such method was developed by Wu and Zhang [218], where they used a smaller number of factors to estimate the covariance matrix for use in a Mahalanobis distance detector to detect

network intrusion. Gauss-Markov Random Fields (GMRFs) have been used to describe the spectral and spatial correlations found in HSI, where the GLRT can be used with parameter estimates of the GMRFs. This is efficient and makes use of spatial information, however, the model assumes that the background is locally homogeneous and has poor performance when targets are located along clutter boundaries [22]. Borghys et al. [34] tried to handle images with more materials in scene by clustering or finding endmembers from a window surrounding each pixel. Each pixel was evaluated by its distance to the cluster centroids, or residual from its endmember linear combination.

Kim and Finkel [123] used LLE and clustering for HSI anomaly detection. First, they used k-means to cluster the pixels, and chose a representative pixel from the center of each cluster. These representative pixels served as the training set for LLE, and the data was reduced to a gray-scale dimension, or three dimensions to be treated as a RGB mapping. The image was mapped by using the cluster assignments and LLE mapping. Zare-Baghbidi and Homayouni [226] split the spectra into two sub-images, and used the maximum and minimum radiance values for each pixel in each sub-image to yield a detector that performed relatively well on a few images.

Billor, Hadi, and Velleman [31] developed the blocked adaptive computationally efficient outlier nominators algorithm (BACON) for identifying outliers in multivariate data. In BACON, an initial basic subset of $m = cp$ exemplars is chosen relative to the entire dataset, where $c = 4$ or 5 typically. Next, two steps are iterated until convergence of the basic subset. First, the Mahalanobis distances are computed using only the basic subset to compute the mean and covariance matrices. Then, a new basic subset is formed by those exemplars with Mahalanobis distance less than $c_{npr}\chi^2_{p,\alpha/N}$, where c_{npr} is a correction factor relative to the variance defined as,

$$c_{npr} = 1 + \frac{p+1}{N-p} + \frac{1}{N-h-p} + \max 0, (h-r)/(h+r), \tag{3.67}$$

and r is the number of exemplars in the current basic subset. h is a parameter that was suggested to be set as $(N + p + 1)/2$, but Smetek [191] suggested $0.75N$ because the contamination level in HSI is usually low. He also suggested $\chi^2_{30,0.0001}$ for the threshold, as otherwise target spectra would be put into the basic subset. The initial basic subset can be chosen using those exemplars with minimal Mahalanobis distance, or those with minimal distance from the coordinate-wise median. The latter proved to yield more robust detection for Smetek [192] in his research with HSI, where he applied BACON to k-means clusters to detect outliers. Béguin and Hulliger [28] used the squared Mahalanobis distance within the BACON framework for their analysis of incomplete survey data.

3.11.6 Means of Identifying, Thresholding, and Comparing Anomalies.

3.11.6.1 Thresholding.

As the images generated by the RX Detector are typically grayscale, Chang [46] developed an automatic thresholding method to find anomalies. A rejection region is defined as $R(\alpha) = \{x|\delta_{RXD}(x) < \alpha\}$ for a grayscale value α and RX score $\delta_{RXD}(x)$, made of all the image's pixels in the RXD-detected image whose gray-level values were less than α. Using a histogram of the detected image, a rejection probability $P(\alpha)$ is defined as $P(\alpha) = Pr(R(\alpha))$. Then a threshold α_0 for detecting anomalies is determined by first setting a confidence coefficient γ such that $P(\alpha_0) = \gamma$, where if $\delta_{RXD}(x) > \alpha_0$ then x is considered an anomaly. This concept is shown in Figure 3.19.

In a similar way, Chiang, Chang, and Ginsberg [55] used a histogram to split pixel scores into background and anomaly classes. Using some bin width determined subjectively, the score corresponding to the first bin of the histogram with no pixels is deemed the threshold value. Clearly, the bin width has to be chosen wisely. This concept is used by both AutoGAD and MPCA for PA SNR and anomaly declaration [107, 110]. An example of this process is depicted in Figure 3.20.

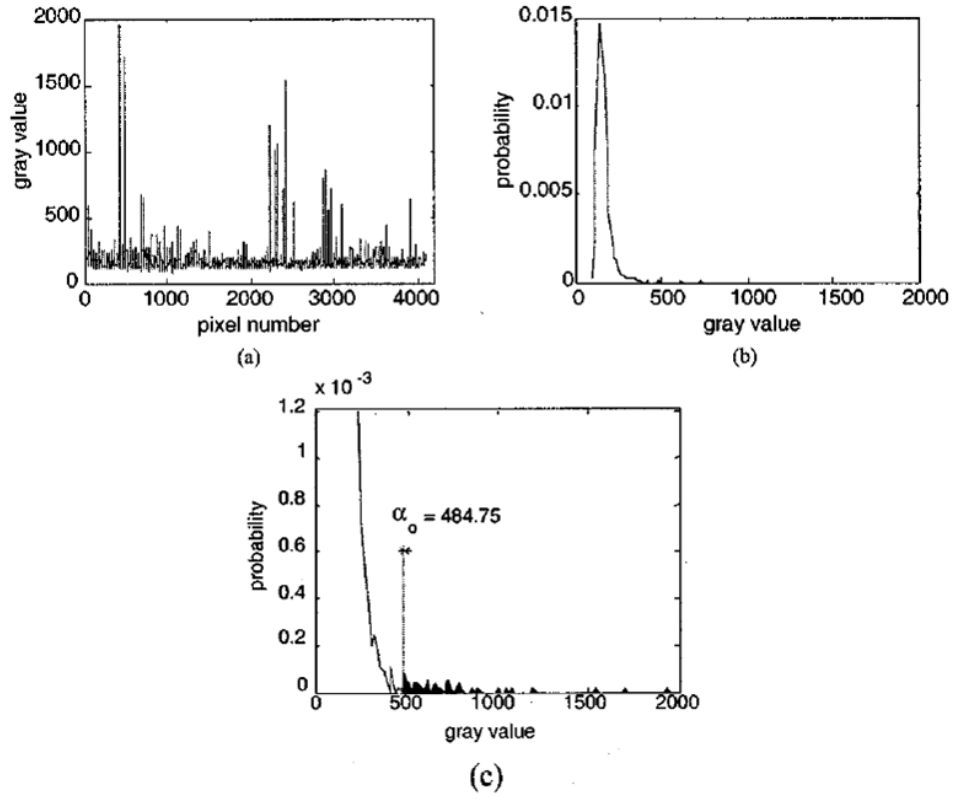


Figure 3.19: (a)Plot of gray values from RXD. (b) Histogram of (a). (c) Enlargement of right tail in (b) [46].

3.11.6.2 Semi-Parametric Test.

Rosario [180] focused on meaningful anomaly detection via indirect comparison between observations. Assuming a local sampling mechanism where two samples are drawn and compared, he determined three study cases as shown in Figure 3.21. The first case is two relatively pure samples from the same population. The second case is two relatively pure samples from belonging to distinct populations. The third case results from a composite or mixture sample, and a single component sample of the mixture. Examples of the first two cases were trees, and a motor vehicle and a grassy area, respectively. To exemplify the third case, he used a sample with two components (*e.g.*, a tree and its

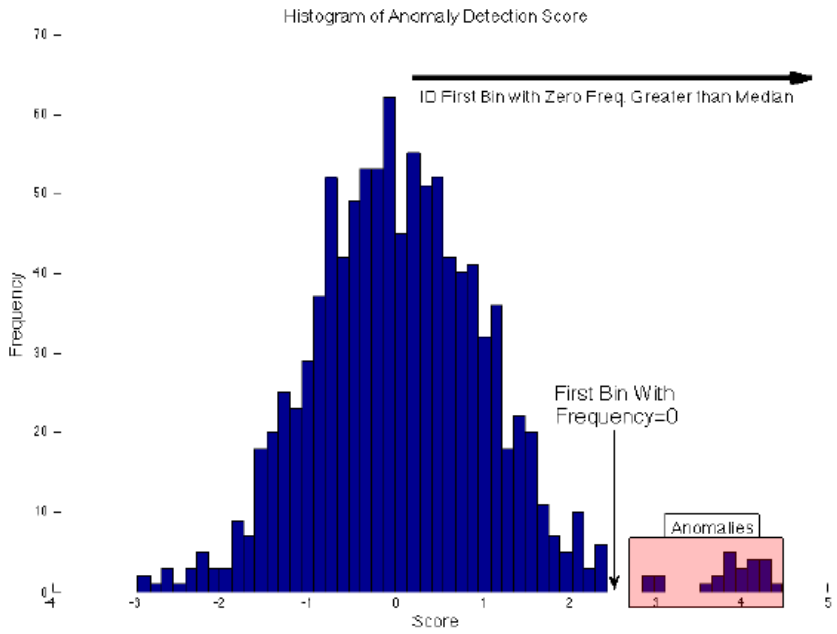


Figure 3.20: Zero-Bin Detection [107].

shadow) and a sample from one the components (*e.g.*, shadow). Rosario [180] noted that this third case appears often when using an inside/outside window mechanism, and that it is responsible for generating a large number of false positives due to abundant region discontinuities in scene imagery. That is, local detectors using conventional statistics often declare a sample of a shadow anomalous relative to a sample with tree and shadow components.

To correct this, Rosario [180] proposed comparing in some form the union of the two samples to one of the individual observations. This would leave the first two cases unaffected in the statistical sense, but would add more evidence about the single component to make the composite a *softer* anomaly with respect to the union of the samples. First, assume two sets of mutually exclusive observations, x_1 and x_2 with respective sizes n_1 and n_2, are available from a general location. Treating these samples as statistically independent

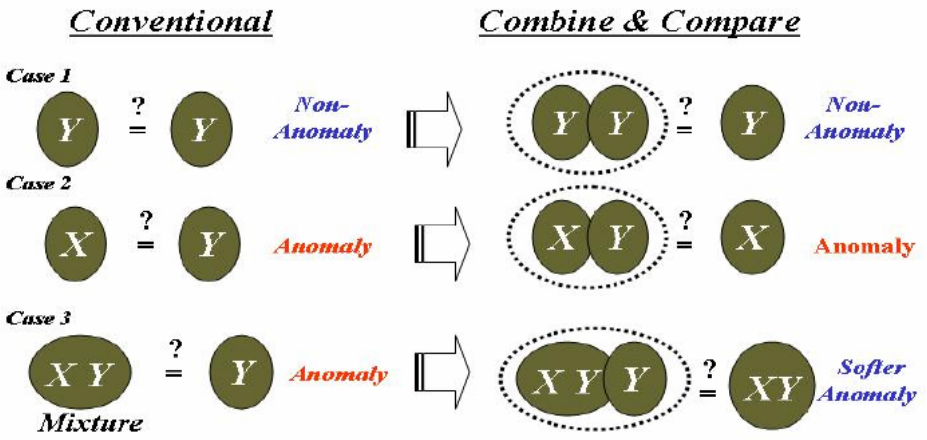


Figure 3.21: Comparison of Samples for Anomaly Detection [180].

random variables with independent and identically distributed (i.i.d.) components, their distributions can be modeled by $x_1 = [x_{11}, \ldots, x_{1n_1}] \sim g_1(x)$ and $x_2 = [x_{21}, \ldots, x_{2n_2}] \sim g_2(x)$. Next, for purposes of building a hypothesis test, these distributions are modeled as $g_2(x) = e^{\alpha+\beta x} g_1(x)$. As α simply acts as a normalizing parameter, this enables the following hypothesis test:

$$\begin{aligned} H_0 &: \beta = 0 \quad (g_2 = g_1) \quad \text{anomaly absent} \\ H_1 &: \beta \neq 0 \quad (g_2 \neq g_1) \quad \text{anomaly present.} \end{aligned} \tag{3.68}$$

After showing the maximum likelihood estimate (MLE) of β, $\hat{\beta}$, was asymptotically Normal, Rosario [180] derived the the following test statistic to test H_0 against χ^2_1,

$$\chi^0 = (n_1 + n_2) \frac{n_2}{n_1} \left(1 + \frac{n_2}{n_1}\right)^{-2} \hat{\beta}^2 \hat{Var}\left([\vec{x_1}, \vec{x_2}]\right), \tag{3.69}$$

where $\hat{Var}$ denotes the variance estimate. Unfortunately, this detector (named SemiP), requires the maximization of a log likelihood function (derived from the likelihood function) to yield the MLEs for $\hat{\alpha}$ and $\hat{\beta}$, and subsequently $\hat{Var}\left([\vec{x_1}, \vec{x_2}]\right)$. This maximization is an unconstrained problem requiring parameter initialization to converge.

3.11.6.3 Non-Parametric F-Distribution Test.

To overcome the limitation of SemiP, Rosario [180] derived a test called the Combined F-Test (CFT) based on the F-distribution and utilizing the *Central Limit Theorem* (CLT). Next, he modeled the random variables x_{ij} as $x_{1j} = \theta_1 + \epsilon_{1j}$ for $j = 1, \ldots, n_1$ and $x_{2k} = \theta_2 + \epsilon_{2k}$ for $k = 1, \ldots, n_2$. Here, it is assumed that $E\left[\epsilon_{ij}\right] = 0$, $Var\left[\epsilon_{ij}\right] = \sigma_i^2 < \infty$, $Cov\left(\epsilon_{ij}, \epsilon_{i'j'}\right) = 0$ unless $i = i'$ and $j = j'$, and the ϵ_{ij} are independent. Letting the combined sample be represented by $y = (y_1, \ldots, y_{n_1+n_2}) = (x_{11}, \ldots, x_{1n_1}, x_{21}, \ldots, x_{2n_2})$, the expected value and variance of the components are $E[y_i] = \theta$ and $Var[y_i] = \sigma^2 < \infty$.

Now defining $\beta_1 = \theta_2 - \theta_1$ and $\beta_2 = \theta - \theta_2$, the hypothesis becomes:

$$H_0 : \begin{array}{c} \beta_1 = \beta_2 = 0 \\ \sigma_1^2 = \sigma_2^2. \end{array} \tag{3.70}$$

Citing his experience with HSI that any sample size greater than 40 satisfies the CLT, he used the *weak law of large numbers* for consistent estimators on the parameters $\left(\theta_1, \theta_2, \theta, \sigma_1^2, \sigma_2^2\right)$ with $\left(\overline{x}_1, \overline{x}_2, \overline{y}, s_1^2, s_2^2\right)$, also yielding $\hat{\beta}_1 = \overline{x}_2 - \overline{x}_1$ and $\hat{\beta}_2 = \overline{y} - \overline{x}_2$ [180]. Using the independence assumptions, and the null hypothesis, two consistent estimators were proposed for the common variance:

$$S_1^2 = \frac{(n_2 - 1)\, s_2^2 + (n_1 - 1)\, s_1^2}{(n_2 - 1) + (n_1 - 1)} \tag{3.71}$$

and

$$S_2^2 = \frac{(n - 1)\, s^2 + (n_2 - 1)\, s_2^2}{(n - 1) + (n_2 - 1)}, \tag{3.72}$$

where $s^2 = (n-1)^{-1} \sum_{k=1}^{n_1+n_2} (y_k - \overline{y})^2$. Utilizing the *Slutsky theorem*, the following test statistic was shown to converge in law to a F_1^1 distribution:

$$Z_{CFT} = \left([n_1 + n_2]^{-1} + n_2^{-1}\right)\left(n_1^{-1} + n_2^{-2}\right)^{-1} \frac{\hat{\beta}_1^2}{\hat{\beta}_2^2} \frac{S_2^2}{S_1^2}. \tag{3.73}$$

Here, rejection of the hypothesis for some chosen type I error indicates x_1 and x_2 are sampled from different distributions, and are thus anomalous to each other. Rosario [180]

showed good results in the univariate anomaly case relative to a basic RX detector when using windows. For these results he applied a high-pass filter in the spectral domain to an inside and outside window to promote statistical independence, formed the first sample using the angle between the outside window filtered samples and the corresponding filtered sample mean, and formed the second sample using the angle between the outside window filtered samples and the corresponding filtered sample mean from the inside window.

3.11.6.4 Spectral Angle Mapper.

One potential issue with finding anomalies is the ability to distinguish between them, perhaps for determination of a soft anomaly. Two basis hyperspectral measures exist for this purpose (although Euclidean distance is another very basic, and as it turns out related, method). Spectral Angle Mapper (SAM) attempts to obtain the angles α formed between a reference spectrum Y and the image spectrum X, treating them as vectors in a space with dimensionality equal to the number of bands [114]. The angle is determined as,

$$\alpha = cos^{-1} \frac{\sum XY}{\sqrt{\sum(X)^2 \sum(Y)^2}} = cos^{-1} \left(\frac{\sum_{l=1}^{p} X_{il} Y_{il}}{||X||_2 ||Y||_2} \right), \tag{3.74}$$

where $cos(\alpha)$ close to 1 represents similarity. Carvalho and Meneses [114] noted the similarity of this metric to Pearson's correlation coefficient, where Pearson's centers the vectors. They showed that SAM is limited by its inability to distinguish between positive and negative correlation, and suggested also centering the vectors as with Pearson's, to suggest positive or negative relationships. The Euclidean Distance between two pixels, that takes into account brightness difference between the two vectors, is relatedly $2\sqrt{1 - cos(SAM(X, Y))}$ [156].

3.11.6.5 Spectral Information Divergence.

Alternatively, Chang [45] developed Spectral Information Divergence (SID) as a criterion. Rather than a geometric focus, each pixel spectrum is treated as a random variable, and the discrepancy of probabilistic behavior between two spectra is measured. As each pixel vector may have unknown interference, the unpredictability caused by the

interference can be described by randomness. First, the pixel vector $\mathbf{x}$ is normalized by the sum of its bands to represent a probability vector q_1. We assume there is another vector $\mathbf{y}$ that has been normalized in a similar fashion to be another probability distribution q_2. SID is defined as,

$$SID(\mathbf{x}, \mathbf{y}) = D(\mathbf{x}||\mathbf{y}) + D(\mathbf{y}||\mathbf{x}), \tag{3.75}$$

where $D(\mathbf{x}||\mathbf{y}) = \sum_{i=1}^{p} q_{1i} \log\left(\frac{q_{1i}}{q_{2i}}\right)$. $D(\mathbf{x}||\mathbf{y})$ is the relative entropy of $\mathbf{y}$ with respect to $\mathbf{x}$, or Kullback-Liebler divergence. Chang developed three discriminatory measures based for any spectral measure m, to include SID. Relative Spectral Discriminality Power (RSDP) compares the discriminality of two vectors s_j and s_k against a third vector x, by,

$$RSDP(s_j, s_k; x) = \max\left\{\frac{m(s_j, x)}{m(s_k, x)}, \frac{m(s_k, x)}{m(s_j, x)}\right\}. \tag{3.76}$$

A higher RSDP implies better discriminality. To investigate the likelihood a signature x is identified by a set of signatures S, the Relative Spectral Discriminality Rate (RSDR) is defined as a probability vector,

$$p_{x,S}(j) = \frac{m(x, s_j)}{\sum_{k=1}^{|S|} m(x, s_k)}. \tag{3.77}$$

To measure the uncertainty of identifying x with respect to the reference signature set, the Relative Spectral Discriminality Entropy (RSDE) is,

$$H_{RSDE}(x; S) = -\sum_{k=1}^{|S|} p_{x;S}(k) log p_{x;S}(k) \tag{3.78}$$

A higher H_{RSDE} implies a lower chance to identify x.

3.12 Image Complexity

Considering the need for proper estimation of background in many anomaly detection algorithms, it would be useful to have a way to classify the complexity of an image, scene, or window. That is, some scenes may have high variability, a sensor may yield some corrupt data, and certain scenes contain many more materials than others. This can all complicate

the ability to remove outliers or fringe anomalies from background estimates. It would also be useful to be able to determine if linear algorithms are sufficient to separate background from anomalies.

In order to assess the non-linearity of pixels, Altmann, Dobigeon, and Tourneret [14] devised a hypothesis test based on estimated source pixels. Assuming the pixel reflectances to be non-linear functions of source spectral components contaminated with white Gaussian noise, they approximated these non-linear functions with polynomials. Specifically, a pixel $\boldsymbol{x}$ can be modeled as,

$$\boldsymbol{x} = g_b(M\boldsymbol{a}) + n, \tag{3.79}$$

where $g_b(\boldsymbol{y}) = \boldsymbol{y} + b\boldsymbol{y}^2$, $M\boldsymbol{a}$ represents a linear mixing of the endmembers in M ny the proportions $\boldsymbol{a}$, and n is the Gaussian noise. To estimate the parameters in a pixel's model, the following problem is solved,

$$\begin{aligned} \min \qquad & J(\boldsymbol{a}, b) = \frac{1}{2}\|\boldsymbol{x} - g_b(M\boldsymbol{a})\|^2 \\ \text{subject to} \qquad & \sum a_i = 1, a_i \geq 0, \end{aligned} \tag{3.80}$$

where the resulting parameter estimators are MLEs.

Nonlinearity detection for each pixel is tested via a hypothesis test on $b \neq 0$. Developing the MLE for $\hat{b}$, they derived a GLRT where $\boldsymbol{a}$ is estimated and the variance of $\hat{b}$ is estimated using a constrained Cramér-Rao bound. Endmembers were found using Vertex Component Analysis. Unfortunately, this test does not directly adapt in terms of determining whether PCA or KPCA is adequate for an image, and there are still parameters to be determined, such as the number of endmembers.

Messinger, et al. [158] used an estimation of the volume of the convex hull enclosing data in the hyperspace to measure image complexity. First, they noted that the determinant of the Gram matrix, the *Gramian*, for any test set is the square of the volume of the parallelepiped formed by the vectors. Breaking the image into small non-overlapping tiles, from each tile they first extracted a large number of estimate endmembers and ordered

them by magnitude. With the linear kernel, they then iteratively estimated the Gramian by adding endmembers. They found this function to reach a peak at a small number of endmembers and then to monotonically decrease. The peak in the Gramian curve was used as the primary means to measure complexity and determine the correct dimensionality of the tile. In this regard, the value could be used to depict the number of materials present in the tile. However, the method relies on good estimation of endmembers and the tile size.

A simple estimate of classification complexity is to use the Fisher ratio, for the two class problem,

$$\sum_{i=1}^{p} \frac{(\mu_{1i} - \mu_{2i})^2}{\sigma_{1i}^2 + \sigma_{2i}^2}. \tag{3.81}$$

Very similar to the Rayleigh coefficient and the F-statistic, one way to extend this to more than two classes is by summing the Fisher scores of each feature [83]. This yields,

$$\sum_{i=1}^{p} \left(\frac{\sum_{j=1}^{c} n_j \left(\mu_{ji} - \mu_{*i}\right)^2}{\sum_{j=1}^{c} n_j \left(\sigma_{ji}\right)^2} \right), \tag{3.82}$$

where n_j is the number of exemplars in class j.

3.13 Receiver Operating Curves

Receiver Operating Characteristics (ROC) is a common method used in anomaly detection to show performance of an algorithm. Given a truth mask, some measure of target detection P_D (True Positive) is plotted on the y-axis against some measure of false detection P_{FA} (False Alarm) on the x-axis to show the trade off as a parameter is varied. To clarify further, a true positive is considered to be when a target pixel is correctly predicted to be such, and a false positive is when a background pixel is predicted to be a target. In this research, True Positive Fraction (TPF) and False Positive Fraction (FPF) are primarily

used for such measures, defined as,

$$\text{TPF} = \frac{\text{Number of True Positives}}{\text{Number of True Positives } + \text{ Number of False Negatives}} = \frac{\text{Number of True Positives}}{\text{Number of Target Pixels}}, \tag{3.83}$$

and

$$\text{FPF} = \frac{\text{Number of False Positives}}{\text{Number of False Positives } + \text{ Number of True Negatives}} = \frac{\text{Number of False Positives}}{\text{Number of Background Pixels}}. \tag{3.84}$$

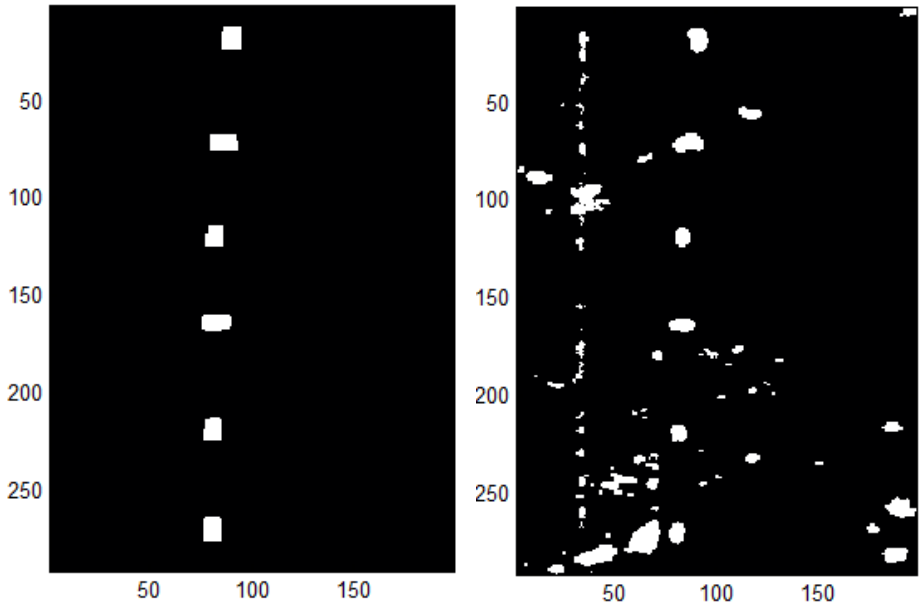
Johnson, Williams, and Bauer [111] also used Label Accuracy (LA) as an additional measure for AutoGAD. They defined this as,

$$\text{LA} = \frac{\text{Number of True Positives}}{\text{Number of True Positives } + \text{ Number of False Positives}}. \tag{3.85}$$

Kwon and Nasrabadi [133] used a slightly different false detection metric when assessing KRX. They defined the false alarm-rate as,

$$N_f = \frac{\text{Number of False Positives}}{\text{Number of Pixels}}. \tag{3.86}$$

Consider an image truth mask and some prediction as depicted in Figure 3.22. Here, there are 57,909 total pixels in the image, with 672 true target pixels and 2,113 predicted target pixels. For this example, TPF is 0.6696, FPF is 0.0291, LA is 0.213, and N_f is 0.0287. The prediction appears like it would have had a higher false alarm measure, but the numbers are so low due to the high number of background pixels and total pixels. This is a consideration that should always be included when using these measures, and it becomes clear that as the number of target pixels increases relative to the size of the image, N_f tends to lower the measure of false alarm. Therefore, TPF and FPF are used as main metrics. Further considerations relative to specific truth masks and metrics are included in Chapter 4.



(a) Truth.

(b) Prediction.

Figure 3.22: Truth and Prediction Example: Targets White.

IV. Investigating Hyperspectral Bands and Truth Masks

Before proceeding to any further analysis on the HSI data sets, it is necessary to analyze their bands and in some cases, their truth masks. The HSI data sets in their unprocessed form still include absorption and noisy bands that should be removed to provide better detection and classification. To do this more rigorously than previously done, a technique is devised to identify these bands. In addition, the majority of the HYDICE images and the HyMAP image have border pixels included in their truth masks, where these are often sub-pixel targets. These have not been consistently treated as background or target necessarily, and so a thorough analysis is done here so as to establish a consistent and correct treatment for this research.

First, a comparison of some similarity metrics is provided in order to justify the choice of similarity used later for similarity-dissimilarity plots. Those plots are then developed and used to show characteristics of the border pixels for the HYDICE and HyMAP images. Finally, the bands of the HSI images are analyzed and a method is developed to identify noisy and absorption bands in an image. These analyses are used to generate the final truth masks and images used in this research, as well as a methodology that can be employed to any image.

4.1 Similarity Metrics

When comparing exemplars, choice of the similarity metric can certainly impact an algorithm. Common distance metrics for natural features include the L_p-norms, defined as,

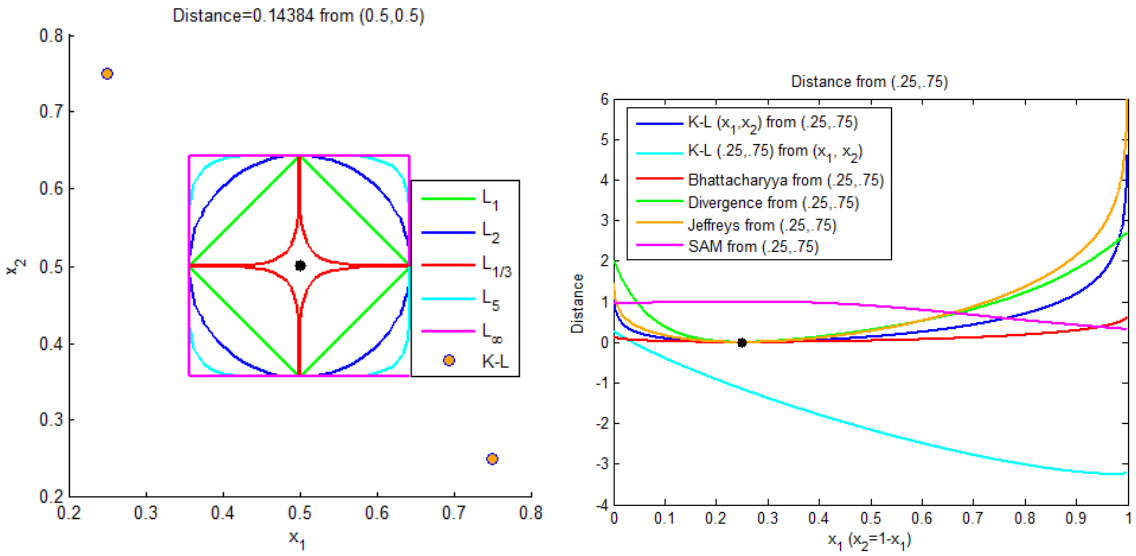
$$L_p(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}. \tag{4.1}$$

Here, d is temporarily used to denote the dimensionality of the dataset and $p > 0$ to represent the exponents, as this is common notation for the L-norm. In the case where

p is infinite, the L-norm is defined as,

$$L_\infty(\boldsymbol{x}, \boldsymbol{y}) = \max_{i=1,\ldots,d}|x_i - y_i|. \tag{4.2}$$

Figure 4.1(a) shows the effect of varying p. The contours shown reflect a constant distance from the point $(0.5, 0.5)$.



(a) L_p-norm Comparison.

(b) Density Distance Comparison.

Figure 4.1: Distance Comparisons.

When the data of interest represents a probability density function (pdf), or can be represented as a pdf, then many additional distance metrics exist. Since all of the pixels in HSI are on the same radiance scale, each element j of a spectral signature can be rescaled as $\frac{x_j}{\sum_{i=1}^{p} x_i}$. This rescaled signature can then be treated as a pdf. In this case, the general shape of the pixel's signature has been unchanged, however, the scales of different pixels relative to one another may now be different.

There are many metrics to quantify the distance between pdfs; a thorough review was given by Cha [43]. Bhattacharyya distance, shown in Equation 4.3, provides bounds on the

Bayes misclassification probability [29].

$$\text{Bhattacharyya}(\boldsymbol{x}, \boldsymbol{y}) = -\ln \sum_{i=1}^{d} \sqrt{x_i y_i} \tag{4.3}$$

Cha [43] showed a level of correlation between L_2 and this Bhattachryya distance. Divergence distance, Equation 4.4, belongs to the squared L_2 family or χ^2 family of metrics [43].

$$\text{Divergence}(\boldsymbol{x}, \boldsymbol{y}) = 2 \sum_{i=1}^{d} \frac{(x_i - y_i)^2}{(x_i + y_i)^2} \tag{4.4}$$

A common metric from the Shannon's entropy family is Kullback-Liebler (K-L) divergence. This is also referred to as relative entropy or information deviation and has many ties with forms of entropy and mutual information [43].

$$\text{K-L}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} x_i \ln \frac{x_i}{y_i} \tag{4.5}$$

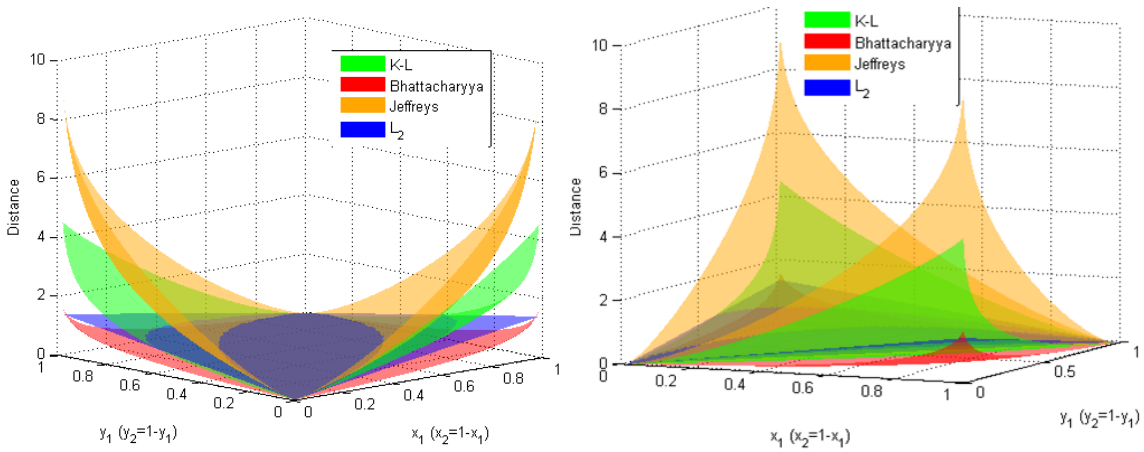
K-L divergence is not symmetric, that is, $\text{K-L}(\boldsymbol{x}, \boldsymbol{y}) \neq \text{K-L}(\boldsymbol{y}, \boldsymbol{x})$. Thus, the value is dependent on which distribution is the reference distribution $\boldsymbol{x}$. A symmetric form developed by adding the two K-L forms is called Jeffreys or J divergence,

$$\text{Jeffreys}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} x_i \ln \frac{x_i}{y_i} + \sum_{i=1}^{d} y_i \ln \frac{y_i}{x_i} = \sum_{i=1}^{d} (x_i - y_i) \ln \frac{x_i}{y_i}. \tag{4.6}$$

These metrics are shown in reference to the point $(0.25, 0.75)$ in Figure 4.1(b), where a two-dimensional probability distribution is varied. Note that fewer points correspond to a single distance in comparison to L-norms. This is exemplified by the two distributions shown in Figure 4.1(a) that have the same distance as the L-norms shown. Bhattacharyya and SAM distance are always between zero and one.

Figure 4.2 further exemplifies the issues of symmetry, and the similarity and correlation of some of these measures. Here, two two-dimensional probability distributions are varied. Note that by varying x_1 for instance, x_2 is automatically defined because $x_1 + x_2 = 1$. It is clear from the figure that differences in the metrics become more

pronounced simultaneously with differences in the distributions. Further, the symmetry issue with K-L divergence is made clear by the different surface direction at the extremes.



(a) (b)

Figure 4.2: PDF Metric Comparisons.

Figure 4.3 compares some of these metrics with Mahalanobis distance, where the covariances $\Sigma_{12} = \Sigma_{21} = 0.75$ and $\Sigma_{12} = \Sigma_{21} = 0.1$ are used for the comparison. Mahalanobis distance has the added advantage that it takes into account the data's covariance structure. Figure 4.4 shows a comparison of a few of the mentioned metrics on three different data sets. Here, the Vertebral Column has negative values, so the data was shifted by a constant before converting to density form for the Jeffreys divergence. In the case of the two HSI images, a random sample of 300 pixels were taken. For each dataset, the similarity between all exemplars was taken, and then the correlation between these similarity value vectors calculated. As expected, the L_p metrics are highly correlated, while Jeffreys and the Mahalanobis distance provide a different look at the similarity. From

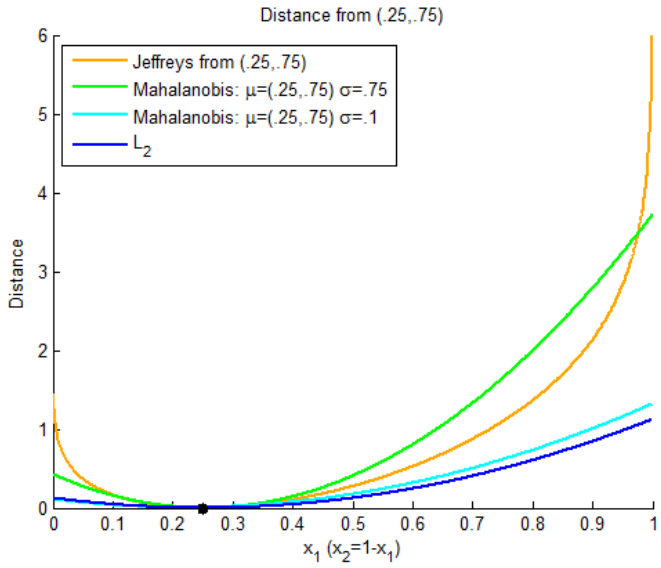
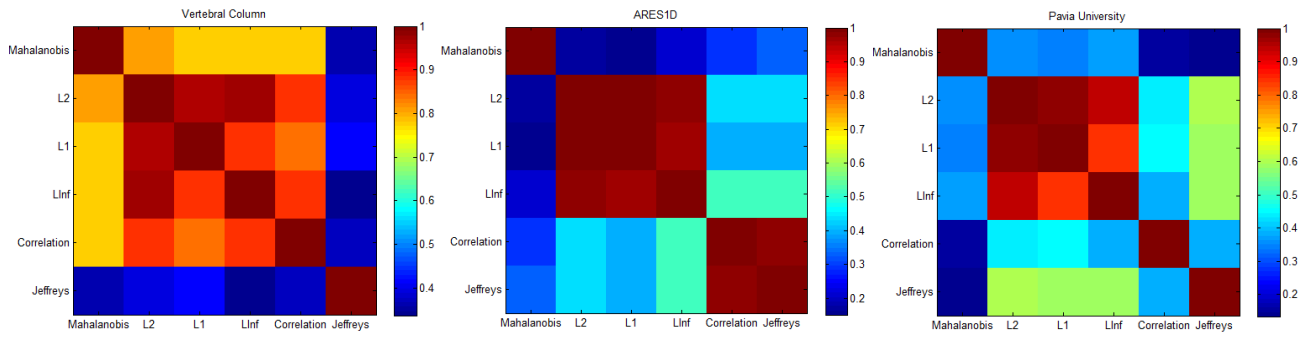


Figure 4.3: Metric Type Comparisons.

algorithms such as RX, the usefulness of the Mahalanobis distance on HSI data has been demonstrated.

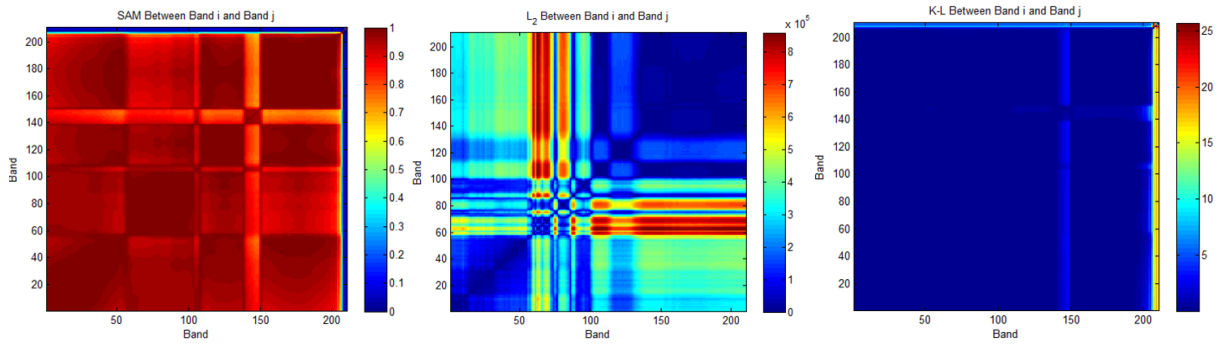


(a) Vertebral Column. (b) ARES1D. (c) Pavia University.

Figure 4.4: Correlation of Similarity Metrics.

Despite these insights into distance metrics, it is still not entirely obvious which metric is best to compare exemplars in any multivariate dataset. However, the author suggests that using L_2 for clustering and similarity comparisons is generally sufficient, and some support is given in the following section beyond what has already been discussed to this point. Use of the L_2 does make an assumption that either all data is on the same scale, or that the features have been normalized, and/or that all of the features are equally important to the analysis. This is entirely true for HSI data, and for when and how the L_2 is used in this research. Any of the metrics that require probability distributions can be problematic, in that the data exemplars (if not already distributions) are changed in magnitude relative to one another. For HSI data, this would be especially troubling, as the radiance magnitudes are important to the characteristics of each pixel. Cha [43] compared metrics using correlation of distances and a clustering dendogram. That analysis suggested moderate relationships between many distance measures, but also showed some level of distinction between L-norm and divergence-based distances. Mahalanobis distance can be useful on HSI data where the background covariance matrix can be used as a tool to better identify anomalies, but even this is problematic. The Mahalanobis distance is not as efficient to compute in comparison to L_p metrics, and it assumes both a good class estimate for the covariance as well as an inherent normality.

Considering again the application of HSI for comparison of bands, Figure 4.5 shows the matrix of distance between all bands using SAM, L_2, and K-L for the ARES4F image. Those bands in the L_2 metric matrix with high values are those where the bulk of the variation in the image occur. The discussed issues are clear in these plots, and were similarly present for all HYDICE imagery. Thus, L_2 is used primarily for similarity in this research as stated previously, although again, the issue of similarity is further investigated as similarity/dissimilarity plots are explored next, followed by visualizations in Chapter 5, and clustering in Chapter 7.



(a) SAM. (b) L_2. (c) K-L.

Figure 4.5: ARES4F Band-by-Band Distance.

4.2 Similarity/Dissimilarity Plots

Arif and Basalamah [15] developed the two-dimensional similarity-dissimilarity plot to reveal class separation, or discrimination quality, in the higher-dimensional feature space. Further, they postulated that it might be used to select an appropriate distance measure for a dataset, with their focus primarily on biomedical data sets.

First, the data is standardized by feature to remove bias using $x_j = \frac{x_j - \mu_j}{\sigma_j}$ for $j = 1, \ldots, p$. Let X denote the entire dataset, X_i^c denote exemplar i of class type c, X^c denote the set of exemplars in X of class type c, and $X \setminus X^c$ denote the set of exemplars in X not of class type c. Then to compute the visualization, the k nearest neighbors of X_i^c in X^c(not including i) and in $X \setminus X^c$ are found. The mean distances of these sets are used as the similarity (comparison to same class) and dissimilarity (comparison to other classes) values, respectively, and are plotted as the axes. This is depicted in Figure 4.6. A line of equal similarity/dissimilarity is plotted as reference, to indicate that points below the line may be difficult to discriminant from other classes as they have a smaller dissimilarity. Data outliers can be spotted on the plot as having large distances.

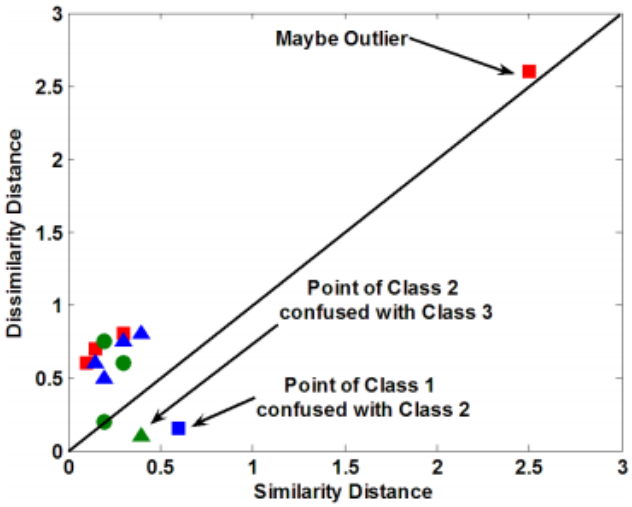


Figure 4.6: Example of Similarity/Dissimilarity Plot [15].

To quantify the expected accuracy of a classifier, the percentage of data points above the similarity-dissimilarity line was suggested as a metric[15]:

$$PAS = \frac{100}{N} \sum_{i=1}^{N} I\{\text{Sim}_{dist}(x_i) < \text{Dissim}_{dist}(x_i)\}. \tag{4.7}$$

This visualization shows much promise in terms of describing classification complexity for a dataset with known classes or for comparing transformations, but it also has inherent limitations. First, a proper distance metric and k need to be chosen for the k nearest neighbors algorithm. In particular, the means could be sensitive to the choice of k for a dataset. Secondly, exact k nearest neighbors can be computationally inefficient for large N. Techniques to help mitigate this with some effect, such as KD-trees, were discussed in Section 3.8.

Figure 4.7 shows the similarity-dissimilarity plot for the Fisher Iris dataset, with $k = 5$ using the L_2 norm. The versicolor and virginica exemplars that are often problematic for classification purposes are evident below the line, and the ease of classification for the setosa class is clear from its PAS of 1. Table 4.1 and Table 4.2 show PAS values for varying

k and similarities for the Breast Cancer and Pima data sets. Some level of insensitivity to k and among L_p norms is present, although this is a small sample and the sensitivity of k could be relative to the size and geometries of the data. It is clear that spectral angles are not appropriate for this natural data, while the probability-based similarities yield PAS values that are not strikingly different. These PAS values also seem to reflect known complexities of these data sets. Figure 4.8 shows a few of the plots for the Breast Cancer dataset.

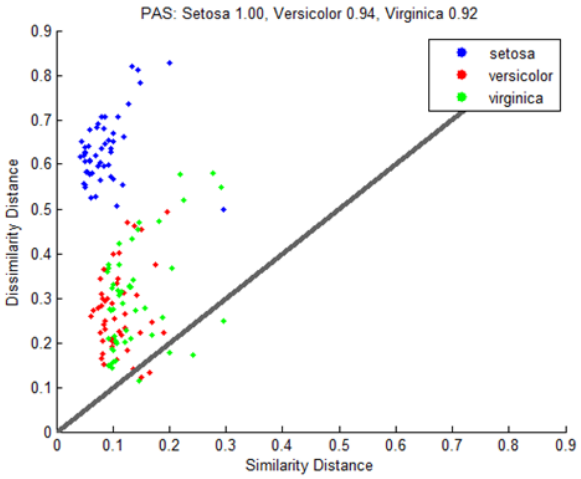


Figure 4.7: Fisher Iris Similarity-Dissimilarity Plot.

Figures 4.9 and 4.10 show similarity-dissimilarity plots for the ARES4F image and image masks for the ratios of similarity values over dissimilarity values. Note, the distances for the SAM metric can be negative, as it is an arc cosine. The author found these ratios to serve as a good alternative view to the similarity-dissimilarity plots. Here, border pixels were also treated as target pixels. A few things can be noticed. First, the L_2 norm provides a better similarity metric in general, even though it is related to SAM. It would be concerning that many of the background pixels have a ratio near one using SAM. This, in conjunction

Table 4.1: Breast Cancer PAS.

Similarity	k	Benign	Malignant
Mahalanobis	5	0.98	0.88
L_∞	5	0.97	0.95
L_1	5	0.97	0.96
L_1	10	0.97	0.95
$L_{1/5}$	5	0.97	0.93
L_2	5	0.97	0.97
L_2	1	0.97	0.91
SAM	5	1	0
SID	5	0.9	0.98
K-L	5	0.87	0.99

Table 4.2: Pima PAS'.

Similarity	k	Negative	Positive
L_2	10	0.84	0.56
L_2	5	0.83	0.57
L_2	1	0.78	0.56
SAM	5	0.91	0.07
SID	5	0.76	0.54
K-L	5	0.77	0.57

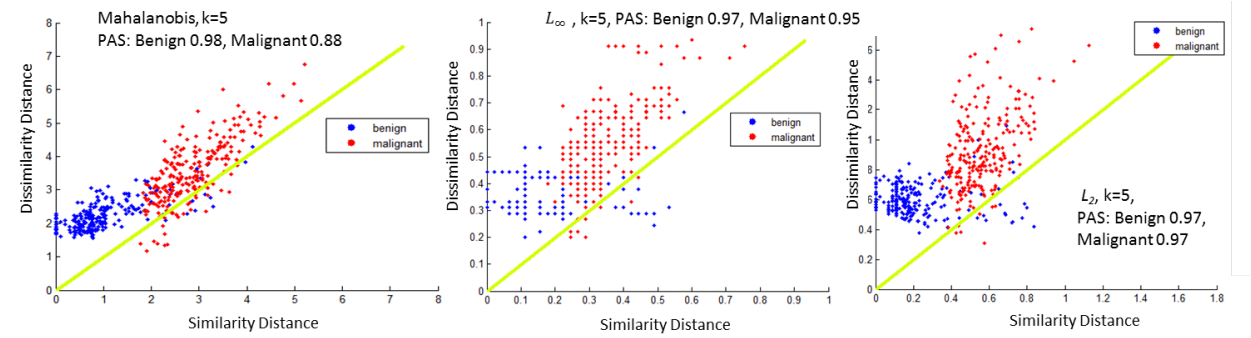
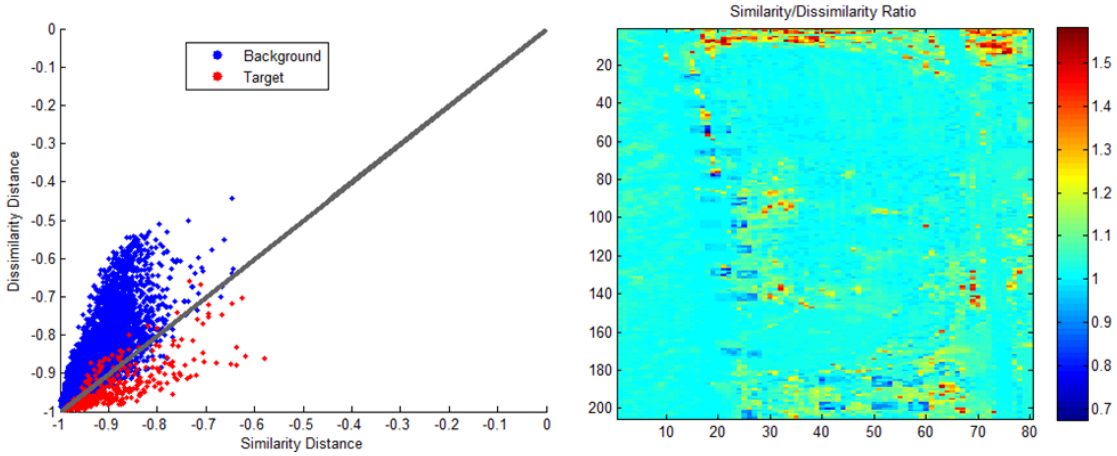


Figure 4.8: Breast Cancer Similarity-Dissimilarity Plots.

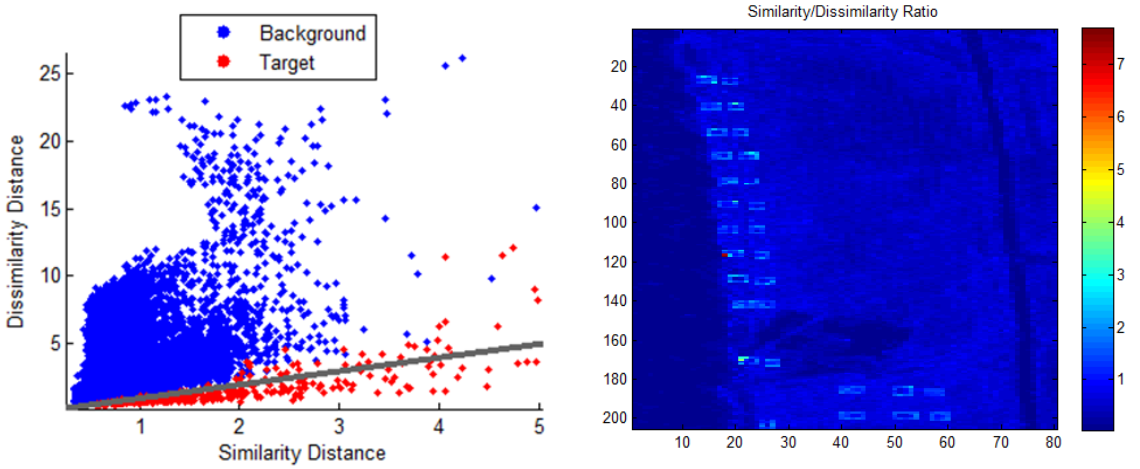
with the previous results suggest that the L_2 norm is a sufficient similarity metric. Second, the border pixels are clearly identifiable in both cases. In the case of the L_2 norm, the border pixels appear to be closer to background then full pixel targets. These trends were consistent across HYDICE images, and so a further analysis was required.



(a) Sim-Dissim.

(b) Sim/Dissim Ratio.

Figure 4.9: Similarity-Dissimilarity Plot - SAM, $k = 5$: ARES4F.



(a) Sim-Dissim.

(b) Sim/Dissim Ratio.

Figure 4.10: Similarity-Dissimilarity Plot - L_2, $k = 5$: ARES4F.

4.3 Analysis of Truth Masks and Border Pixels

As was shown in Table 2.1, many of the HYDICE images have pixels classified as border, and in some cases these outnumber the full-pixel targets. Even after comparing pixel signatures to nearby full-pixel targets and background pixels, it is not obvious how to treat these pixels. Additionally, the HyMAP image has pixels classified as guard pixels accompanied by the same issues.

4.3.1 HYDICE.

One immediate use of the Similarity/Dissimilarity plots is to be able to better investigate the truth masks for the HYDICE-derived and HyMAP imagery (and ROSIS and AVIRIS imagery). In previous research using the HYDICE imagery [74, 107, 110], based on archived code and run analysis, these border pixels were treated favorably. That is, border pixels predicted as anomalous were included in True Positives, but border pixels were not included in False Negatives. This boosted TPF values. Similarly, border pixels were included in True Negatives if not classified as anomalous. This decreased FPF values.

Treatment of the border pixels in this manner is highly favorable, and it is not obvious how these pixels really should be treated so as to provide fair ROC curves. Therefore, a more rigorous analysis is required.

First, let us compare the Fisher ratios (Equation 3.81) for background, border, and target pixels, shown in Table 4.3. With the exception of ARES3F, these numbers indicate that the border pixels are further separated from the target pixels than the background pixels. Table 4.4 shows the ratios when the border pixels are treated as background and when they are treated as target pixels. Again, in all cases except ARES3F, these metrics seem to indicate the border pixels may truly be more like the background than any significant sub-pixel targets. In the case of ARES3F, the Fisher ratios are near equal in both cases.

Table 4.3: HYDICE ARES Fisher Ratios.

Image	Background/Target	Background/Border	Target/Border
ARES1D	161.09	36.31	263.56
ARES2D	9.32	0.41	9.45
ARES1F	54.45	1.85	60.86
ARES2F	186.49	35.93	168.70
ARES3F	50.30	66.24	30.08
ARES4F	10.95	6.91	16.43

Now, removing the noisy and absorption bands from these HYDICE images as determined by Smetek [191] and used by Johnson [110] (145 bands remaining), using the L_2 metric, and $k = 5$, the PAS values are explored. Table 4.5 shows the PAS values treating each image as two-class, where background is compared against combined target

Table 4.4: Modified HYDICE Fisher Ratios.

Image	Border as Background	Border as Target
ARES1D	161.59	6.56
ARES2D	9.32	1.43
ARES1F	54.54	13.42
ARES2F	185.07	52.18
ARES3F	49.67	51.45
ARES4F	10.96	4.25

and border, target by itself, border by itself, and border and target are compared. These values again suggest that the border pixels are generally more similar to background pixels than target pixels, and here also in the case of ARES3F. Admittedly, these are at a global level, but attempting to generate PAS values for local windows to also incorporate spatial information (comparing background, target, and border inside of windows like in RX) is problematic in that those windows would not always contain different classes of pixel or to have anything from another class relatively nearby. This global evaluation is very much relevant anyways, as the methods evaluated in this research are largely global detectors and methods.

Figure 4.11 shows the similarity-dissimilarity plots for ARES3F. As can be seen, the distribution of the distances are relevant to the analysis as a high PAS could have many dissimilarities that are only slightly larger than similarities, and as seen in these cases, the scales of the distances are also important to consider. Figure 4.12 shows the plot when combining the target and border pixels to treat as a target class. The histograms, where the bin counts shown are the maximum bin count, make it clear that many of the pixels are near the line. These various plots together, which was common amongst images, indicate that

Table 4.5: HYDICE PAS Values.

Class/Image	ARES1D	ARES2D	ARES1F	ARES2F	ARES3F	ARES4F
Background	0.999	0.994	0.999	0.999	0.999	0.999
Target/Border	0.388	0.261	0.559	0.204	0.253	0.165
Background	0.999	0.999	1.000	1.000	0.999	1.000
Target	0.919	0.901	0.935	0.788	0.703	0.404
Background	0.999	0.995	0.999	0.999	0.999	0.999
Border	0.073	0.074	0.129	0.047	0.013	0.038
Border	0.966	0.996	0.928	0.994	0.984	0.976
Target	0.940	0.904	0.950	0.824	0.897	0.587

there is no clear distinction between the border pixels and some of the background pixels. Additionally, there is a higher level of distinction between many of the border pixels and many of the target pixels.

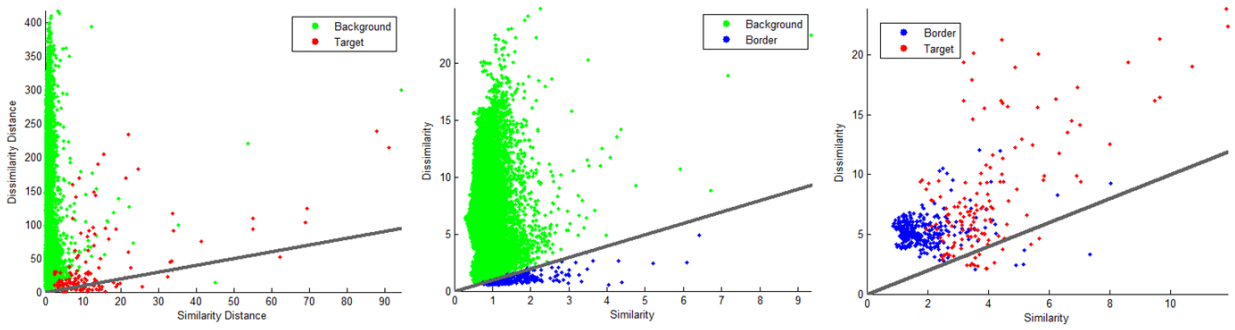


Figure 4.11: ARES3F Similarity-Dissimilarity Plots.

To further exemplify these points, a plot based on unstandardized data and a plot with $k = 10$ are shown for ARES1D in Figure 4.13. There is little change when not standardizing

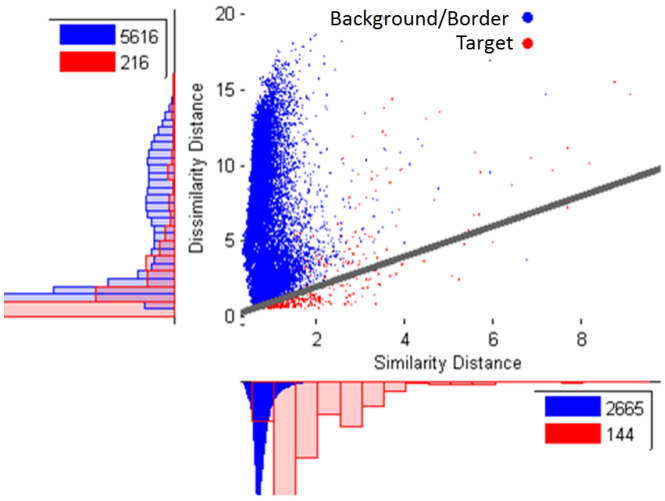


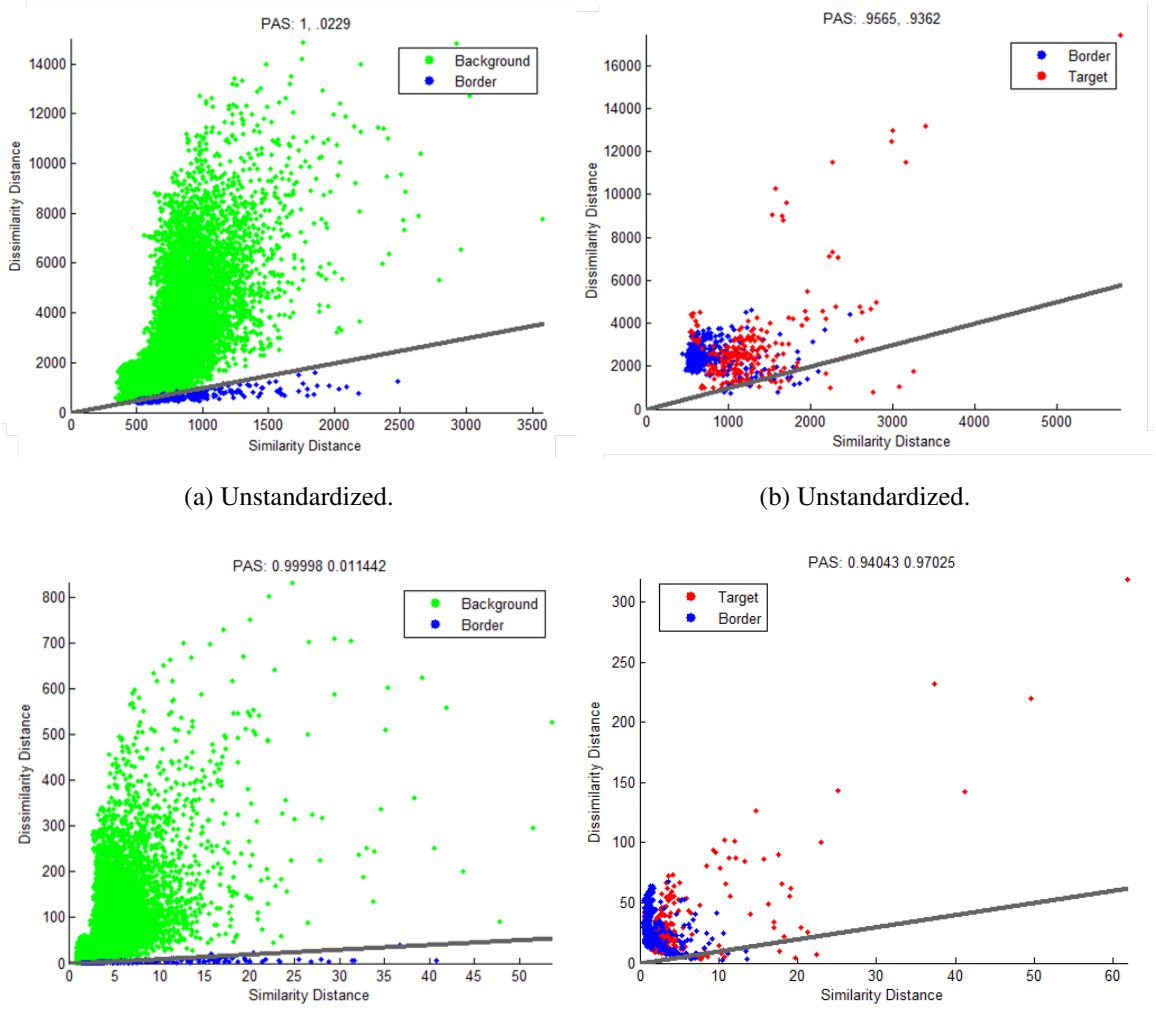
Figure 4.12: ARES3F Plot: Target and Border vs. Background.

due to HSI bands already being on the same scale, and a larger k did not have substantial effect. Also, the divide between target and border pixels is more pronounced here, as would be expected based on the Fisher ratios.

In the cases where not all of the exemplars belong to the two classes in the visualization (*i.e.*, three of the four cases from Table 4.5), the different data standardizations make it such that the dissimilarity distances are not directly comparable. Thus, to make this comparison fairly and even more extensively, the data can be standardized based on all exemplars and then the dissimilarities can be computed for the border pixels. This enables a plot and Percent Closer to Background (PCB) metric analogous to the similarity-dissimilarity concept, to be defined as,

$$PCB = \frac{100}{N_{Border}} \sum_{i=1}^{N_{Border}} I\{\text{Dissim to Background}_{dist}(x_i) < \text{Dissim to Target}_{dist}(x_i)\}. \quad (4.8)$$

A table of PCB values for the ARES images is shown as Table 4.6 for $k = 5$ and $k = 20$, where now all non-zero bands are used for the computations. Again, these indicate that the border pixels are more dissimilar to the target pixels than the background pixels.



(a) Unstandardized.

(b) Unstandardized.

(c) $k = 10$.

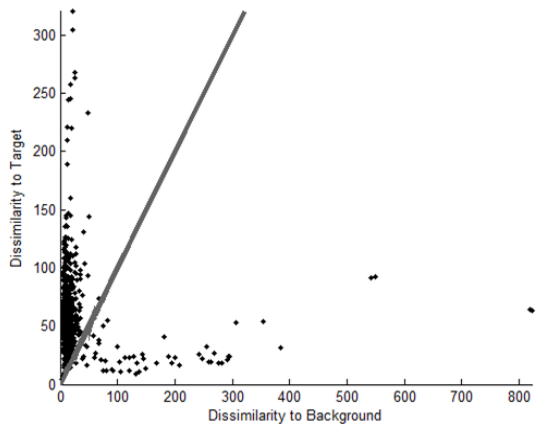
(d) $k = 10$.

Figure 4.13: ARES1D Similarity-Dissimilarity Plots.

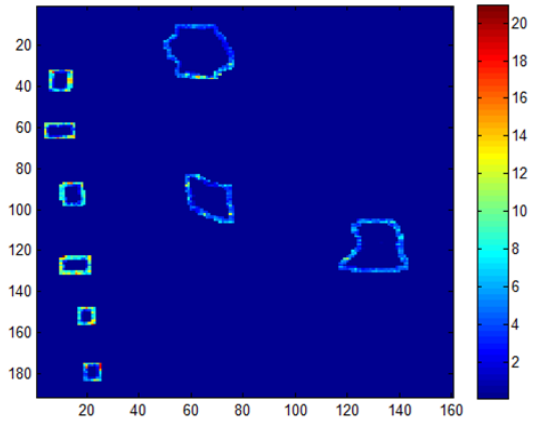
Figure 4.14 depicts the associated dissimilarities and dissimilarity ratios (border-target over border-background) for ARES2D and ARES1F. These had the largest outliers of all images by far in terms of a point with high dissimilarity to background. It is clear that most of the border pixels are significantly closer to background pixels. This is particularly interesting in the case of ARES2D, due to the number of targets in the image. The ratio heat plots also indicated no clear trend relative to the locations of the border pixels.

Table 4.6: HYDICE PCB Values.

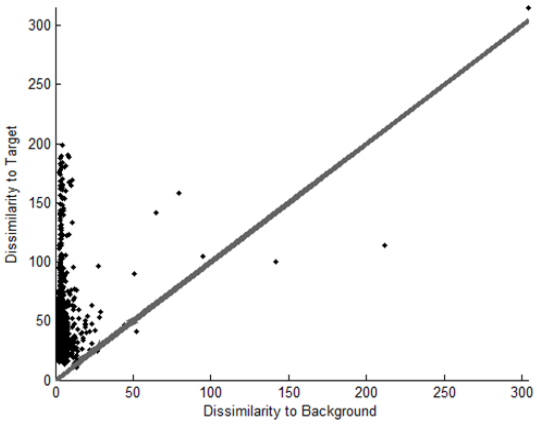
k	ARES1D	ARES2D	ARES1F	ARES2F	ARES3F	ARES4F
5	0.986	0.997	0.939	0.991	0.997	0.994
20	1.000	0.999	0.941	0.996	1.000	1.000



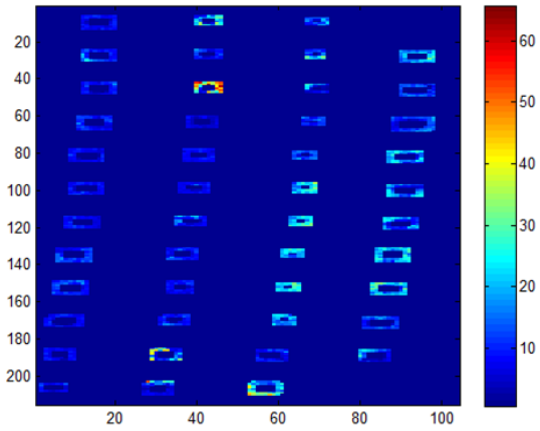
(a) ARES1F Dissims.



(b) ARES1F Dissim Ratio.



(c) ARES2D Dissims.



(d) ARES2D Dissim Ratio.

Figure 4.14: Border Pixel Dissimilarities with $k = 5$.

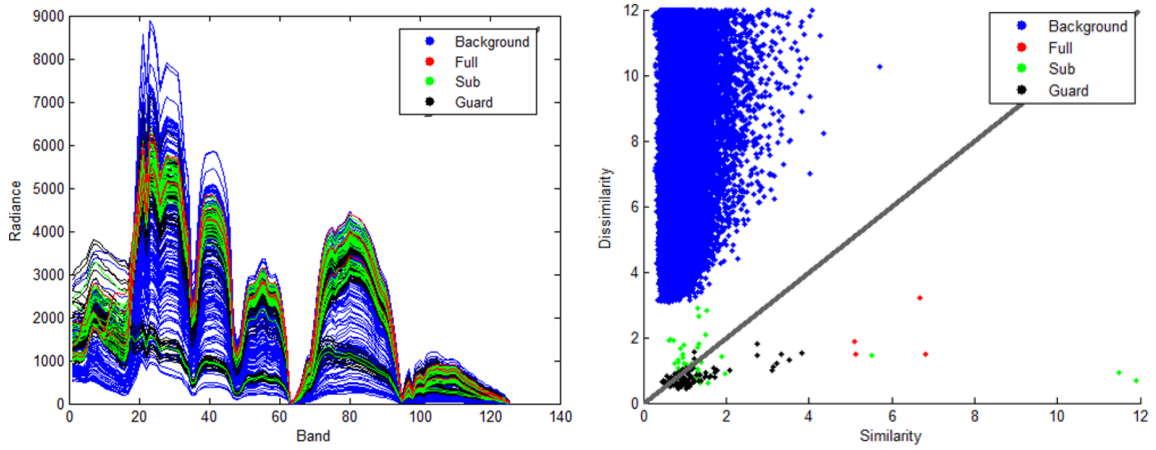
These analyses indicate a lack of clear break between the border pixels and background pixels. Perhaps more importantly, they indicate that including the border pixels in True Positives may in fact be ignoring False Positives. The absolute sub-pixel make-up of these pixels is impossible to determine. Therefore, for the remainder of this research these border pixels are treated as background for TPF and FPF measures, vice what has been done in the past in the literature, unless otherwise denoted. This makes the ROC curves and measures more conservative, treats the border pixels consistently, and places a desired emphasis on finding the full-pixel targets. However, the percent of border pixels declared anomalous is also measured. This enables conversion to previous measures if ever necessary, as TPFs are potentially lower and FPFs higher as a result of this decision. Additionally, the number of targets detected including the full-pixel targets only, and then including border pixels is also used as a set of two measures. These are described in full in Section 4.4.

The run03m20 image has no border pixels. The PAS for the background is 0.960 and for the targets is 0.403. The latter value and the accompanying similarity-dissimilarity ratio plot indicated the possibility that certain target pixels are similar in nature to the border pixels from the ARES set. However, no effort is made to adjust the truth mask here as there were no border pixels in the original mask, and this dataset can be thought of as a case where the border pixels are treated as targets instead of background.

4.3.2 HyMAP.

The HyMAP image also has border pixels, and in fact, only has four full-pixel target pixels. In this case, the 141 border pixels are broken out into sub-pixel and guard categories in the truth mask. Treating these as targets, and comparing targets to background, the PAS value for background is 0.999 and for targets is then 0.207. In this case the Fisher ratio is also 57.94, whereas if the border pixels are treated as background, the Fisher ratio becomes 182.29. Figure 4.15(a) shows the signatures of these pixels and 200 random background

pixels from the image. Clearly both the full-pixel targets and border pixels are difficult to distinguish from background at the global level. Figure 4.15(b) shows a portion of the similarity-dissimilarity plot for this data with $k = 5$ and the L_2 metric, where the three non-background pixel types were compared individually against all other classes. This further clarifies that although these pixels are close to background, in some cases they are closer in nature to one another than any background pixel.



(a) HyMAP Signatures.

(b) HyMAP Sim-Dissim Plot.

Figure 4.15: HyMAP Truth Mask Analysis.

Despite the findings here, this image still has extremely few target pixels to detect even with inclusion of the border pixels. Therefore, for purposes of this research the border pixels are still treated as targets.

4.3.3 AVIRIS & Pavia.

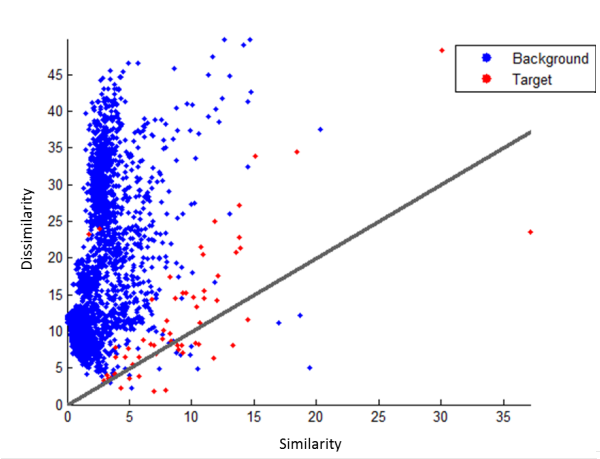
The remaining HSI images in this research either have no border pixels or no truth masks. Removing 45 bands from the AVIRIS images for this analysis, due to covering areas of known absorption [2], the AVIRIS target classes all have a PAS between 0.7 and

0.85 with $k = 5$ and the L_2 norm, and background PAS of 0.999. This indicates decent separation of target from background. Their respective two-class Fisher ratios are: 4Ship2 with 79.40, Scene1 with 84.26, Ship1 with 21.30, and VirginIslands1 with 235.58. Figure 4.16 depicts the similarity-dissimilarity plots for these images and reflect these findings. In some cases, the axes were scaled so as to show the majority of the data because a few pixels have extremely high similarity and dissimilarity distances. As a result of this analysis, the truth masks for these images are modified in no way for this research.

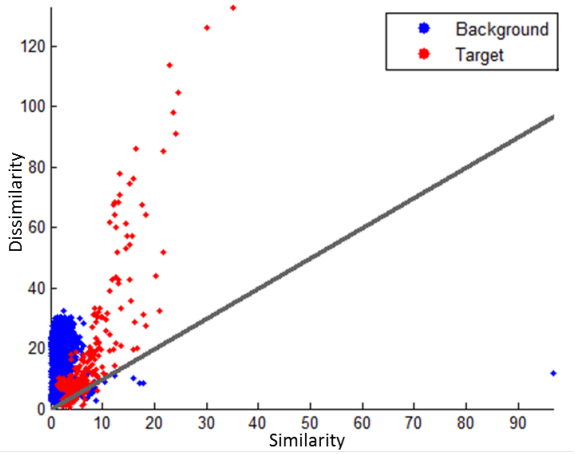
The Pavia (or ROSIS) data sets have no anomaly-background truth masks. Instead, they have 10-class material truth masks. Table 4.7 shows the PAS values with L_2 norm for these classes in the Pavia University image, where each class was compared against all other classes. This reveals that the Painted Metal Sheets class could potentially be treated as an anomalous class. A plot of the similarity-dissimilarity ratios, Figure 4.17, reveals pixels that may be more prone to being detected as anomalous. Of course, this would depend on the scale of dissimilarity relative to what a given algorithm considers anomalous.

4.4 Additional ROC Metrics

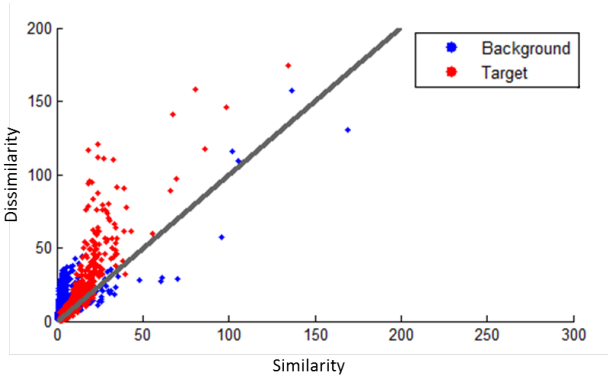
Given the pixel analysis for the various images with truth data, metrics, in addition to TPF and FPF, seemed suitable to provide better analysis. Again, in the TPF and FPF metrics for this research, the border pixels for each image set are treated as just discussed. Although these choices on how to treat the border, sub, and guard pixels for the various image sets have been defended, the tracking of certain metrics that use the original truth masks also make the impact of these decisions on the TPF and FPF more obvious. This is important because although it has been shown that the border pixels are closer to background, that is not to say that there are no sub-pixel targets or mixed pixels. Further, the new metrics provide an alternative view of the success of any detector. The following three metrics are



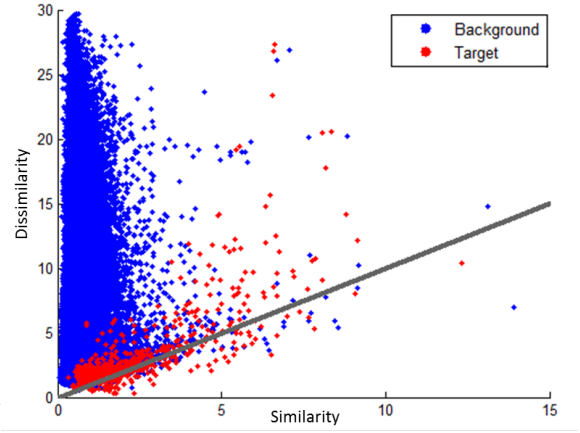
(a) VirginIslands1.



(b) 4Ships2.



(c) Scene1.



(d) Ship1.

Figure 4.16: AVIRIS Similarity-Dissimilarity: $k = 5$.

proposed:

$$PTNB = \frac{\text{Number Targets Detected (Not Including Border Pixels)}}{\text{Number of Targets}}, \tag{4.9}$$

$$PTIB = \frac{\text{Number Targets Detected (Including Border Pixels)}}{\text{Number of Targets}}, \tag{4.10}$$

$$PBDA = \frac{\text{Number Border Pixels Declared Anomalous}}{\text{Number of Border Pixels}}. \tag{4.11}$$

Table 4.7: Pavia University PAS Values.

Class	$k = 5$	$k = 25$	$k = 50$
Background	0.948	0.967	0.972
Meadows	0.642	0.598	0.553
Asphalt	0.433	0.329	0.293
Bare Soil	0.308	0.122	0.080
Self-Blocking Bricks	0.459	0.399	0.352
Trees	0.286	0.128	0.082
Gravel	0.458	0.245	0.151
Painted Metal Sheets	0.848	0.906	0.914
Bitumen	0.683	0.712	0.684
Shadows	0.387	0.187	0.124

The Percent Target No Border metric (PTNB) tracks what percentage of targets were detected, using only the full-pixel or non-border target pixels. The Percent Target Including Border metric (PTIB) yields a similar metric, but also allows the border pixels (and guard and sub pixels for HyMAP) to be treated as target pixels. Both of these metrics enable an alternative success measure that evaluates if all targets are detected vice just a large percentage of target pixels. The third metric, Number Border Pixels Declared Anomalous (PBDA), is a measure that directly enables other treatments of TPF and FPF based on how one wishes to view the border pixels. Further, all of these are metrics can be used as part of a ROC curve to evaluate parameters.

4.5 Hyperspectral Band Selection and Analysis

Band selection is a research area in and of itself for HSI, and can even be an alternative to dimension reduction [47]. When using PCA and other dimension reduction techniques,

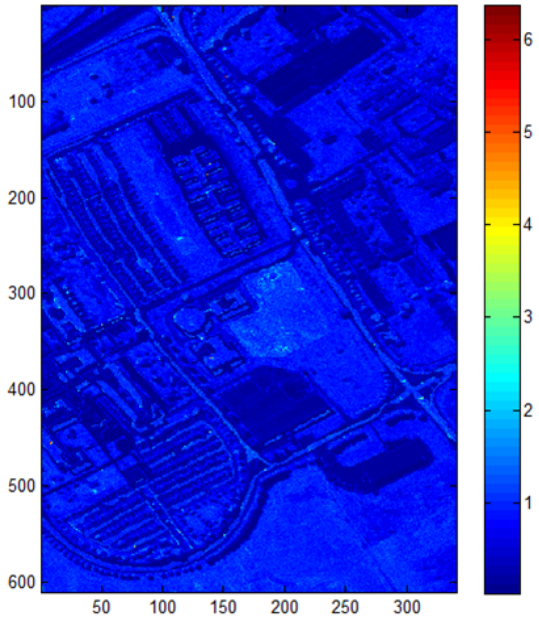


Figure 4.17: Pavia Univ Similarity/Dissimilarity Ratio: $k = 5$.

the objective is often to try and find some underlying structure that can simplify the problem. Such techniques may also reveal bands that contain little information useful towards the designated purpose, or data redundancy. With or without pre-processing, the images may still contain absorption or noisy bands as well, that provide little benefit.

A common approach is to identify and remove absorption/noisy bands based on properties of the spectrum, possible materials in the scene, and properties of the sensor before applying any technique to the data. The intent of such a 'supervised' approach is to ensure that these bands do not contribute any unnecessary complexity. However, there appears to be no consensus on how to actually perform this identification, and it is not uncommon to see different bands removed on the same images in the literature. Additionally, there appears to be a misconception between actual absorption bands, and bands that appear to be noisy due to error in collection or other factors. Examples of this with the HYDICE ARES images are provided shortly in Section 4.5.1. Ideally, any

methodology used would remove absorption bands, remove noisy bands that would be of little value to any anomaly detection algorithm, and not be flexible to different sensors and areas of the EM spectrum. Similar concepts can be applied to any dataset with features to remove noisy or low-information features. To demonstrate this, the technique developed here is applied to the Arcene dataset as well in Section 4.5.6.

Lavanya and Sanjeevi [136] used factor analysis and correlation analysis to evaluate bands with little discriminatory information for a hyperspectral image of an agricultural area. Their process is depicted in Figure 4.18. First, correlations between bands were computed, and subsets of bands with correlation above 0.8 were removed. From the correlation matrix of the retained bands, principal component-based factor analysis was performed, keeping only two factors. The resulting high loadings were compared with other metrics to determine a final band set. Importantly, known water absorption bands were removed before any of their analysis. The technique developed here also leverages factor analysis, but does not assume the location of absorption bands are known.

Pu and Gong [173] summed the squared PCA loadings for a varying number of eigenvectors, selecting those bands with highest value for retention. Miller [160] used correlation in an attempt to reduce the number of bands fed into ICA during the AutoGAD algorithm, after absorption bands had already been removed. Noting that neighboring bands often had correlated information, he defined clusters of bands by using a correlation threshold and comparing bands in sequence until the threshold was not met. This breakpoint indicated the start of a new cluster. Each cluster could then be used to define a representative band. Cai [42] noted that neighboring absorption bands might have low correlation and used a threshold to identify them. Chang [47] developed a series of methods primarily based on constrained energy minimization (CEM) to select bands, finding a general correlation of the band to the image. However, some of these required computation

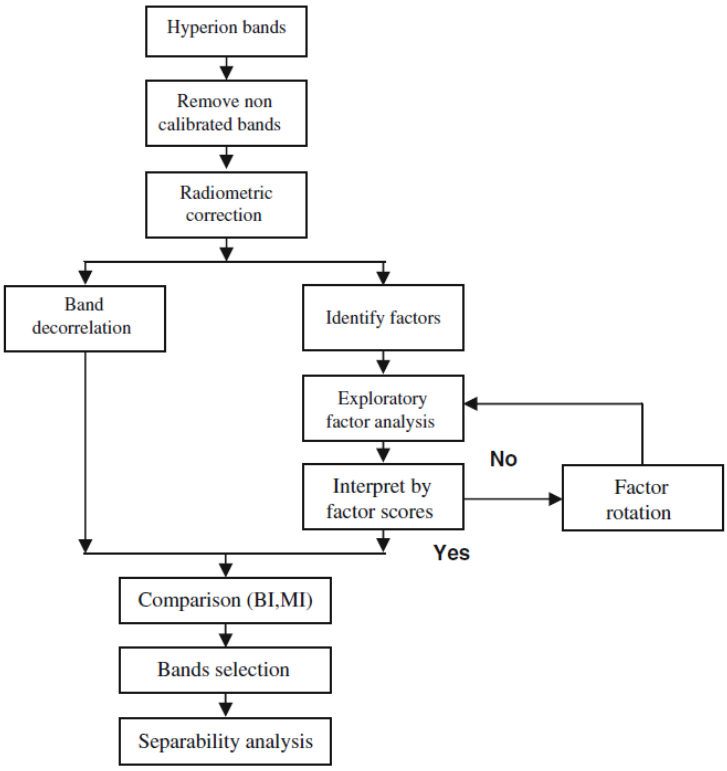


Figure 4.18: Band Selection Methodology [136].

and inversion of the $N \times N$ pixel correlation matrix, while the others developed to avoid such a large computation yielded an entirely different band selection.

Martinez-Uso, et al. [155] clustered on the bands using mutual information between bands and Jeffreys divergence between bands. The mutual information between two bands was defined as the sum of entropy for each band minus their joint entropy. Jeffreys divergence was as defined in Section 4.1. Their use of mutual information had the added complexity of needing to build a histogram in order to yield the joint entropy for the bands. Also note that to use entropy on the bands, they must be normalized by the sum of elements so as to reflect a probability distribution. This changes the relative magnitude of pixels to one another, emphasizing the shape of the signature. Datta, et al. [63] selected bands in a process finding highest correlated partitioned bands, and removing those with largest K-L

distance from a Gaussian image. This process is problematic as the Gaussian assumption does not always hold in HSI [76].

Kesheva [120] used the Spectral Angle Mapper to find bands that would best discriminate between known materials. Sotoca, et al. [196] devised a criterion named the Minimization of the Dependent Information (MDI), based on entropy, as a means to measure the dependency of a set of bands. The measure, the joint entropy of a set minus the conditional entropies for the bands in the set, could be used to determine the information gain by adding bands in a forward approach. Unfortunately, the method was designed for multi-spectral images where the number of bands is much lower and calculating the MDI is not intractable. Wang, et al. [210] used a spatial correlation to choose bands, but used class information in aid of this spatial mutual information. Zeng and Durrani [227] used copulas to calculate the mutual information of bands and to provide a band selection procedure. Efficiency of such a technique could be an issue however, as the copula function has to be integrated or summed over all values of each band distribution, *i.e.*, the pixels. Zare [225] developed a method to simultaneously estimate endmembers and perform band selections using sparsity promoting priors. This approach involves a quadratic program, however, and has tunable parameters.

Faulconbridge, Pickering, and Ryan [71] developed an unsupervised method in an attempt to yield an automation that would remove bands that provide little aid to classification. First, they normalized the data cube for an AVIRIS image (a different image than those in this research) using the image mean and standard deviation. Next, k-means with $k = 10$ and the L_2 norm was used to cluster the pixels. In order to quantify the separation between clusters in each band vice the entire set of bands, a modified Bhattacharyaa distance was used,

$$B_{ij,p} = \frac{\left(C_{i,p} - C_{j,p}\right)^2}{4\left(\sigma_{i,p}^2 + \sigma_{j,p}^2\right)} + \frac{1}{2}\ln\left(\frac{\sigma_{i,p}^2 + \sigma_{j,p}^2}{2\sigma_{i,p}\sigma_{j,p}}\right), \tag{4.12}$$

where $B_{i,j,p}$ is the distance between cluster i and cluster j at band p, $C_{i,p}$ is the centroid of cluster i at band p, and $\sigma_{i,p}$ is the standard deviation of cluster i at band p. As this distance is by band, they then used a threshold on the maximum distance between any two clusters to nominate bands for removal where there was little statistical separation. This technique was problematic in that there was no obvious way to determine an appropriate threshold. Nonetheless, later this method without the threshold is used to aid validation of the algorithm developed in retaining the correct bands.

None of the methods in the literature seem to be a truly robust way to identify absorption, noisy, and low-information bands in the unsupervised setting. Yet, the concepts have been shown to be useful, and might be used in some fashion so as to explore and identify bands for removal. To reiterate, this removal serves as a pre-processing step before employing any anomaly detection algorithm. Removal of the low-information bands may still leave bands that contain much of the same information, but if these bands aid in anomaly detection then the redundancies present are not necessarily a negative quality. A possible method to be quickly investigated shortly is the use of signal-to-noise ratio (SNR).

Before proceeding, it is necessary to calculate a table of known weak and strong absorption locations for the various sensor types, given in Table 4.8, to have for comparison. These locations are calculated exactly based on the spectrum ranges, number of bands, and information and sources previously presented in Section 2.3.1.1. As an example, if the sensor collected on the spectrum from 0.39 to 1 μm over 128 bands, the band location for 0.6 μm is approximately,

$$\frac{0.6 - 0.39}{(1 - 0.39)/128} = 44.07. \tag{4.13}$$

Locations are rounded to the nearest integer, and in some cases, not all of the mentioned absorption spectrum locations occur within the sensor's collected range. It should be mentioned that the absorption is likely in surrounding bands as well. In the cases of ROSIS and SpecTIR, images in this research have a slightly different number of bands

(103 and 102, 356 and 360), so ranges are given when appropriate to cover all images. Also, the Red Sea image is made distinct from other SpecTIR images as its bands cover a different range of the spectrum from the others. As a rule of thumb, absorption increases from weakest to strongest, to some extent, as μm increases.

Table 4.8: Absorption Band Number Locations.

μm	HYDICE	AVIRIS	ROSIS	SpecTIR(Not Red Sea)	Red Sea	HyMAP
0.600	20	21	40 : 41	36 : 37	44	9
0.660	26	28	55	47	57	13
0.730	33	35	71 : 72	59	71	17
0.820	42	45	93	74 : 75	90	23
0.910	51	54		89, 91	109	28
0.940	54	58		95 : 96	115	30
1.140	74	79		130 : 131		42
1.375	98	104		170, 172		57
1.900	150	160		261, 264		89
2.500	210	224				126

Before developing a methodology fully, it must also be considered that not every image has an entire set of valid bands. In other words, due to collection error or very strong absorption a band may consist entirely of zeros or have a large portion of the image that has zero radiance recorded. These bands are of no use, and in the case where a large portion of the band is zero, can confuse the factor analysis-based method that is developed here. The variance presented by having many zeros and some non-zero pixels can sometimes be misinterpreted as informational by an algorithm. Table 4.9 shows the bands for HYDICE and AVIRIS images that have more than 50% zero pixels (or erroneous pixels that have

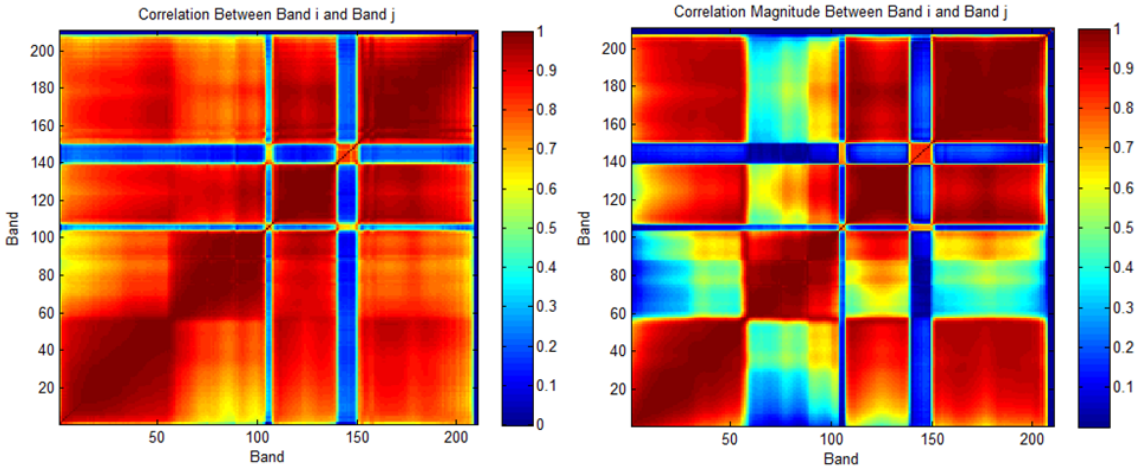
negative value and were set to zero). No bands have this characteristic for the images from other sensors. It can be seen that these bands are all very near, or are, the "exact" locations of absorption from Table 4.8. 50% was chosen as the threshold both observationally based on the data sets in this research, and because it makes sense operationally that any band with 50% zero pixels has a large amount of noise or error.

Table 4.9: Bands with > 50% Zero Pixels.

Image	Bands
ARES1F	$141:149, 207:210$
ARES2F	$104:108, 139:151, 207:210$
ARES4F	$207:210$
ARES1D	$142:149, 209:210$
ARES3F	$208:210$
ARES2D	$104:108, 139:152, 208:210$
ARES1C	$147:148, 206:210$
ARES2C	$140:150, 200, 202:210$
run03m20	$104:108, 139:152, 208:210$
4Ships2	$154:168, 223:224$
Scene1	$108:110, 153:169, 219:224$
Ship1	$153:168, 223:224$
VirginIslands1	$107:115, 152:169, 218, 220:224$

Figure 4.19 shows the correlation magnitude between bands for images ARES1D and ARES4F. As can be seen, some of the strong absorption areas including those with many zero pixels are obvious due to extremely low correlation. Additionally, some of the weaker areas are clear by the blocking present. This suggests that covariance or correlation could be

a key to finding the noisy and absorption bands as is developed shortly. This also displays the limitations of some of the correlation methods currently in the literature. Neighboring sets of bands can still sometimes be highly correlated to one another, making a neighbor approach prone to missing noise or absorption if noise happens to be highly correlated to its neighbors, despite the presence of the block effect. This suggests the need for a more global model without a neighbor pre-processing.



(a) ARES1D. (b) ARES4F.

Figure 4.19: Band-by-Band Correlation Magnitude.

4.5.1 Band Selection Method and HYDICE-Derived Data.

In order to help develop and justify the technique presented, the HYDICE imagery is used as an example. The HYDICE images have 210 bands between 0.397 and 2.5 μm. For the ARES set of images, Smetek [110, 191] devised that the 145 bands 5:72, 78:85, 92:99, 116:134, and 158:199 should be used, as he deemed the others to be absorption or too noisy based primarily on very low values [30]. This set of bands became a standard used by Johnson [110], Jablonski [107], and many others for their respective algorithms.

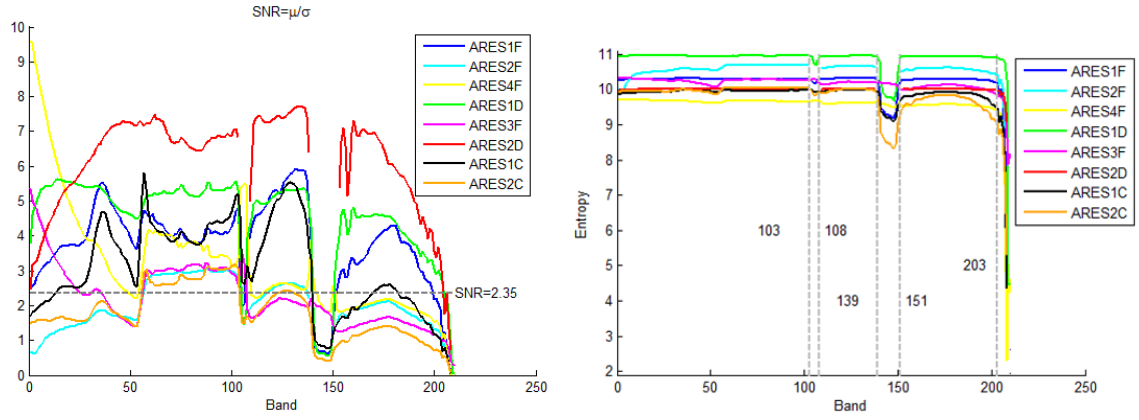
Interestingly, Friesen [74] used a slightly different set of retained bands for these HYDICE images: 10:97, 115:132, and 158:200. Bihl, et al. [30] identified 104 bands: 18, 20, 24:25, 27:28, 30, 38:61, 65, 66, 69:73, 78:85, 93:102, 116:133, 162:184, 188:191, 194:196, and 199. To make this determination they used, in part, signal-to-noise ratio, windowing, and neighboring band correlation and they took into consideration a specific algorithm's performance. Kwon and Nasrabadi [133] used a fourth set on HYDICE images different than the images in this research (one of these images was very similar to ARES1D): 23:101, 109:136, and 152:194. Further, Miller [160] noted that bands 73:77 and 87:91 showed no obvious indication of noise or lack of information on the ARES images, indicating a set closer to that of Kwon and Nasrabadi [133]. These slight variations are likely not of huge concern, but they also serve as an opportunity to investigate a methodology to find absorption and noisy bands, other than simply looking at the pixel radiance in each band visually and making a subjective decision.

We can first consider $SNR_p = \mu_p/\sigma_p$, where μ_p and σ_p are the mean value and standard deviation band p, as a simple way to detect low-information bands. The ratio of the mean to the standard deviation of the pixel signatures is an estimate for the image SNR, although it is often an underestimate due to interpixel variability [157]. Absorption causes signatures to drop and noise yields variability, and so variability of a band, by itself, is not necessarily a good metric. As seen in Figure 4.20(a), the use of any threshold on this SNR varies by characteristics of the image and it is not obvious how to adjust a threshold. For instance, variations on using the standard deviation and range of bands to commonly scale the SNR did not provide a consistent SNR measure. This is not surprising, as the materials and abundance of materials in an image varies the relation of the mean to the variance. Further, only the high absorption bands become obvious by using this technique, as evidenced by sharp drops in the SNR in bands 140:150 (near 1.9 μm) and 203:210 (near 2.45 μm). Entropy is another approach to try and determine the information content of each

band. If each band is made into a vector $\boldsymbol{b}$ and normalized to a probability-like vector such that $\sum_{i=1}^{N} b_i = 1$, then the entropy of the band can be calculated as,

$$-\sum_{i=1}^{N} b_i \log\left(b_i\right). \tag{4.14}$$

Using entropy as a direct measure of band information, again only the latter two strong absorption areas are obvious. However, as shown in Figure 4.20(b), the third major absorption area of the spectrum near 1.375 μm is noticeable. Although these techniques may identify absorption bands, literature suggests that there are other areas of absorption or noise that they do not identify.



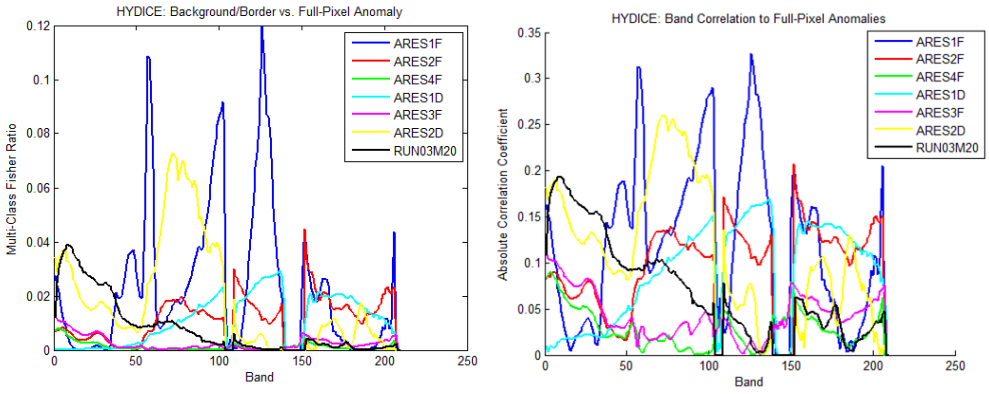
(a) SNR.

(b) Entropy.

Figure 4.20: Simplistic Band Selection.

Before proceeding, it is useful to try and further evaluate the bands of the HYDICE images to see if there is an 'ideal' set of bands to maintain, especially given the variation in those used within literature. Figure 4.21 depicts the multi-class Fisher ratios and correlations to full-pixel targets or anomalies for each band, for the HYDICE images with anomalies in the image. Here, this correlation was calculated between radiance values and the truth mask. There are no consistent trends, aside from the clear absorption near band

150. Figure 4.22 shows the maximum $B_{i,j,p}$ (Equation 4.12) for each band. The problem with thresholding this metric, as mentioned previously, is clear from this plot. This also provides no consistency across images, aside from confirming an area of absorption near band 150 and a possible area of noise after band 100. Certain band sets do appear to be better discriminators for specific images, however. In conclusion, for HYDICE these metrics do not really help identify a set of bands desirable to retain across all images.



(a) Multi-Class Fisher Ratio. (b) Correlation to Full-Pixel Anomalies.

Figure 4.21: HYDICE Band Metrics.

Now, recall that band signatures tend to group on the spectrum. That is, subsets of the bands are usually highly correlated, as was shown back in Figure 2.9. Miller [160] used this fact to find these correlated segments and select a subset of bands accordingly. This correlation structure might also be utilized to search for absorption and noisy bands by using Factor Analysis. Using MDSL to select the number of factors for the model, Figure 4.23 depicts the specific variance of each band for each ARES image using two variants of factors. Here, 0.8 specific variance on the y-axis denotes 80% and the MDSL threshold was adjusted by -1 to reflect the setting from AutoGAD.

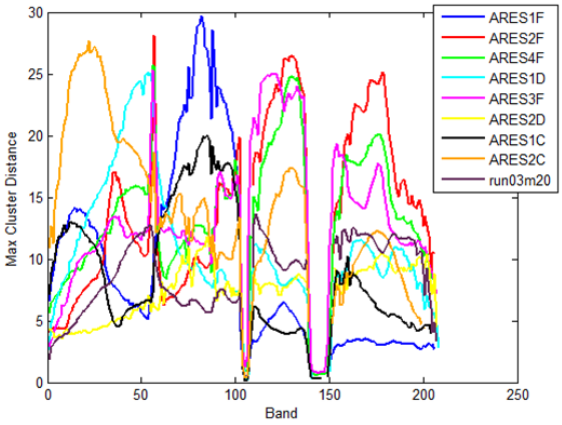
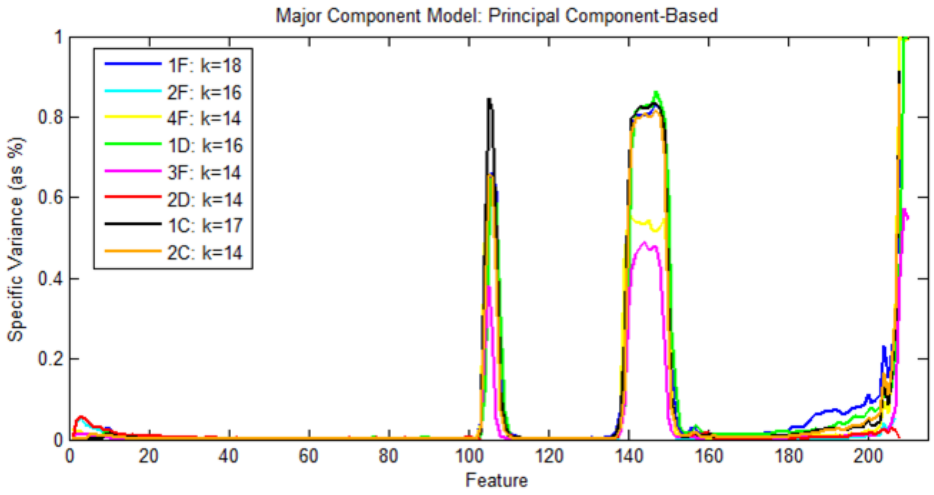


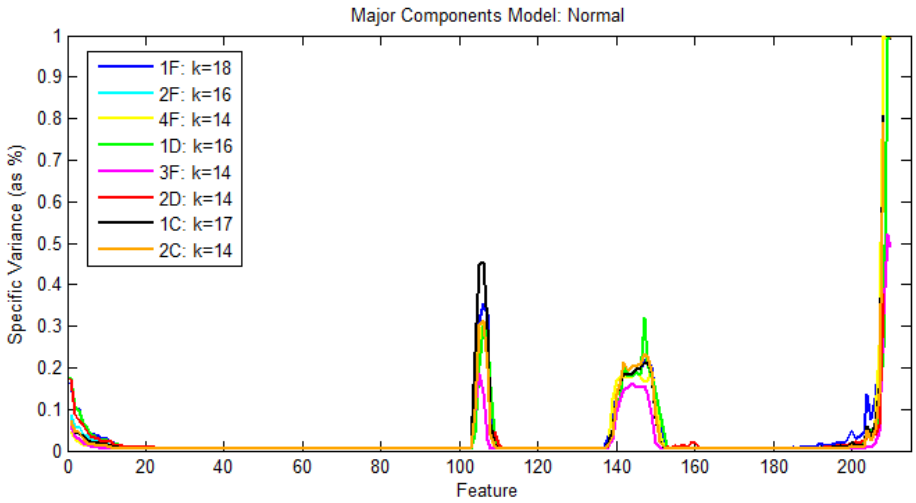
Figure 4.22: HYDICE: $\max_{i,j} B_{i,j,p}$.

The three major areas of absorption bands are clear in the specific variance plots, corresponding to high specific variance, although they are more pronounced in the PC-based model. The bands removed by Smetek and Miller resemble an aggregate across images where the specific variance rises. It is evident that a threshold may be able to identify absorption and noisy bands. Additionally, from these plots it would seem that a threshold could also be flexible to different image characteristics.

The factor analyses are based on the leading k_0 factors of the centered data, where k_0 is determined using MDSL. Thus, it is important to assess the sensitivity of the specific variances to k_0, *i.e.*, does a small change in k_0 significantly affect which bands are retained given a constant threshold. Figure 4.24 shows the effect of varying the number of factors on the specific variances for ARES1D and ARES2F, as examples due to their differences. The sensitivities in these plots are representative of all of the HYDICE images, where some images displayed even less sensitivity. As can be seen, the specific variances are fairly robust to changes in k_0. However, they do change enough that as the change in k_0 increases, so too does the change in which bands are retained. Of course, this change in retention does not become pronounced (> 10) until the change in k_0 exceeds 5, in which case the



(a) PC-Based Factor Analysis.

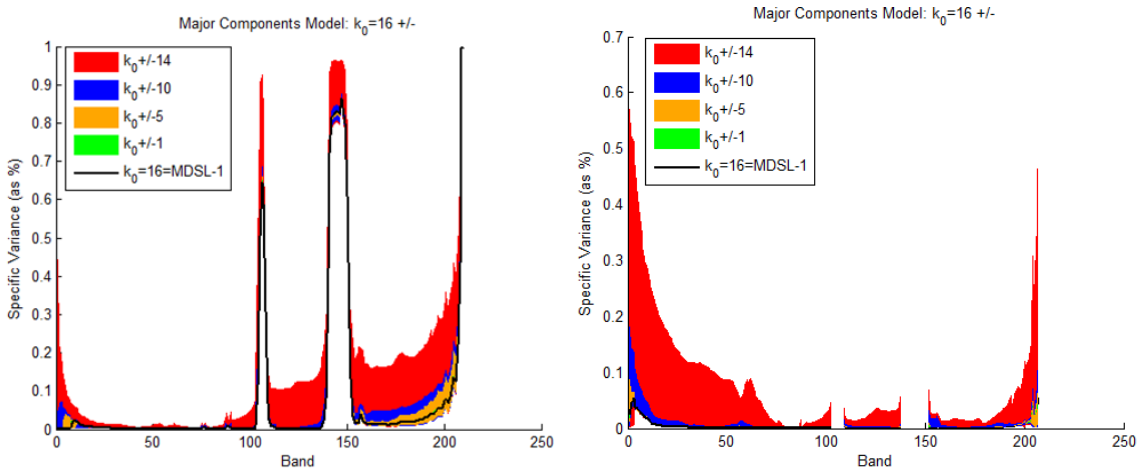


(b) Normal-Based Factor Analysis.

Figure 4.23: Specific Variance: ARES Images.

factor models are likely becoming too large or too small. Thus, whereas AutoGAD uses a dimension adjustment parameter in the MDSL algorithm, no adjustment to the MDSL cutoff is proposed and used from here on for simplicity.

Based on these observations and experimentation, the process shown in Figure 4.25 was developed. After reshaping the image, the largely-zero bands are removed so as to not



(a) ARES1D. (b) ARES2F.

Figure 4.24: Specific Variance: MDSL Effect.

provide false variance into the model. That is, bands with many zeros and some high values could mistakenly be retained if not removed *a priori*. Next, a factor model is built using a number of factors as determined by the MDSL algorithm. After a varimax rotation so as to yield high loadings and group the bands best onto factors, the specific variance is used to identify bands that are highly noisy or that provide little to the model. Admittedly, the varimax rotation is an unnecessary step because the specific variances are rotation invariant. However, it is included such that the resulting factors can also be interpretable.

The remaining issue is how to determine a valid threshold. Figure 4.26 shows the number of images in which each band would be removed, given a threshold. Clearly, the bands to remove are not entirely consistent across all images, aside from the main absorption areas. This makes sense when also looking at the individual bands of the images. Given this information and those bands removed in the literature, the threshold of interest lies below 0.05. In fact, experimentation indicated that a threshold of $t = 0.02$ was ideal. Therefore, if $\psi_i \geq 0.02$, band i is removed.

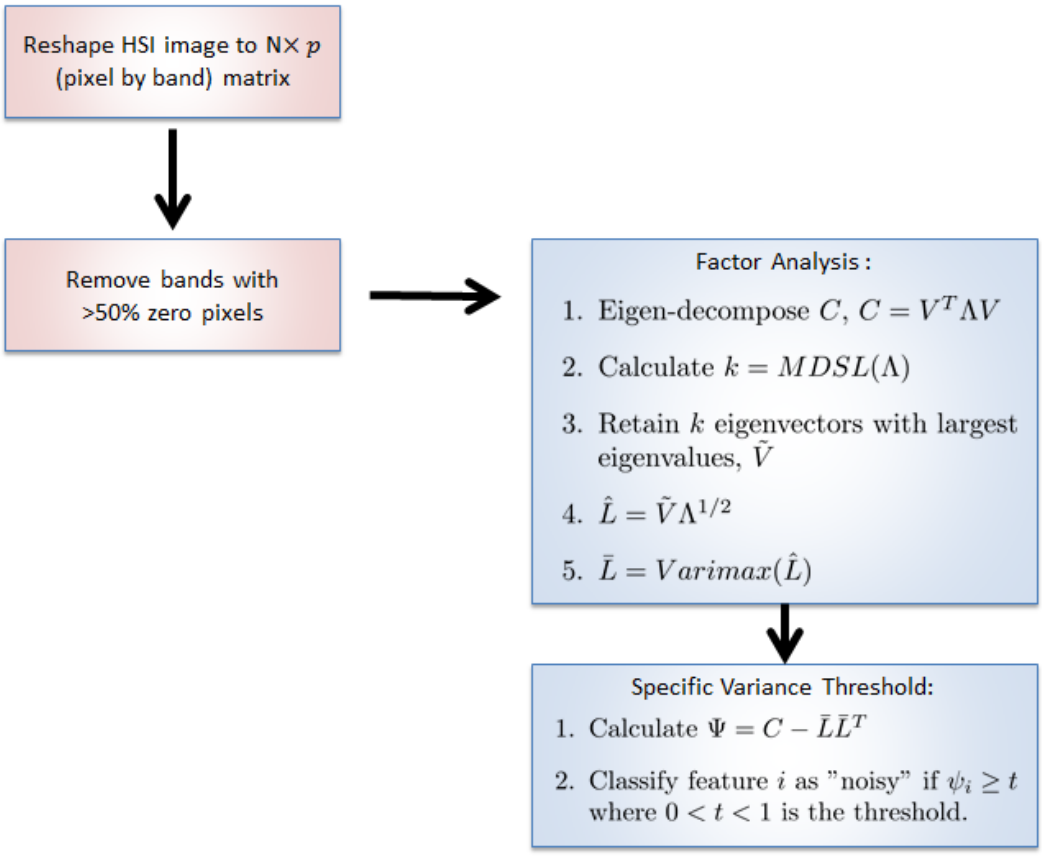


Figure 4.25: Band Selection Methodology.

This is supported by Figure 4.27. Here, the colors denote the band being removed if that threshold was used. That is, the red bands would be removed for all three thresholds in each plot, and blue would be removed for the blue and green thresholds. Across all images, a threshold of 0.02 eliminates bands in all three high absorption areas of the spectrum (the latter three exact absorption locations). Additionally, on an aggregate level, this threshold covers much of what was determined to be noisy in the literature. It can also be seen that this method allows flexibility to the image as not all images are noisy in the same areas of the spectrum, which matches what occurs within the images.

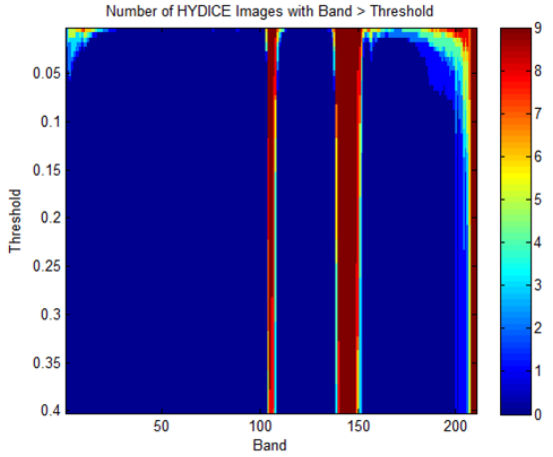


Figure 4.26: HYDICE Images: Threshold Sensitivity.

In reality, the exact threshold, similar to the literature, can become a subjective decision as to how high radiance values need to be in order to deem a band noise. In some cases, it appeared that bands were removed in the literature due to the presence of very slight artifacts in the band, even though the band is arguably not noisy. Here, 0.02 was chosen because it removed unquestionably noisy bands, retained bands that distinguished objects well within the radiance levels, and covered the locations from literature very well. For example, Figure 4.28(a) shows a band from ARES1D typically removed in the literature that is no longer removed, whereas Figure 4.28(b) shows a band from ARES1F that is typically retained that is now removed. The radiance values are extremely low in the ARES1F band, while the ARES1D band has reasonable radiance values and distinguished objects. The true criticality of retaining certain bands or removing them is lessened by the fact that several neighboring bands that were retained/removed in the literature, in comparison to any opposites here, are highly correlated share some of the same feature information.

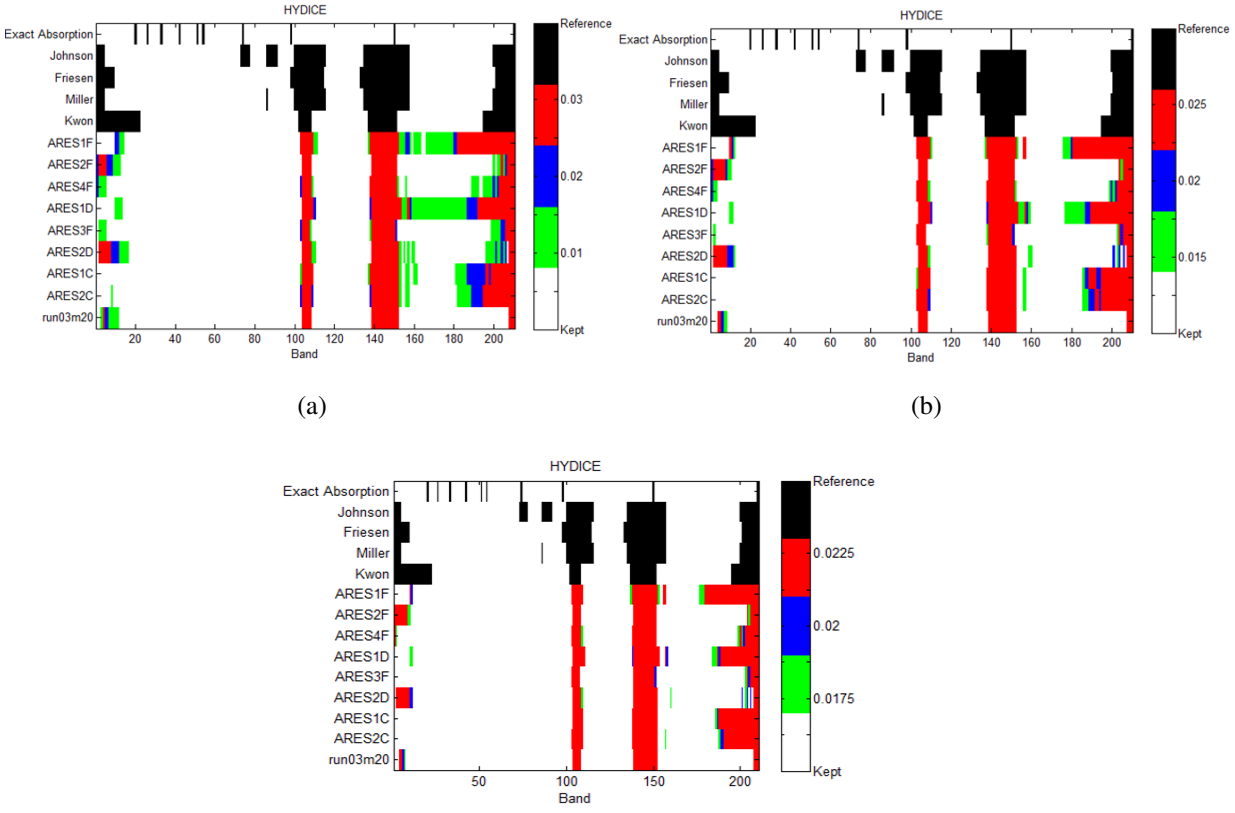
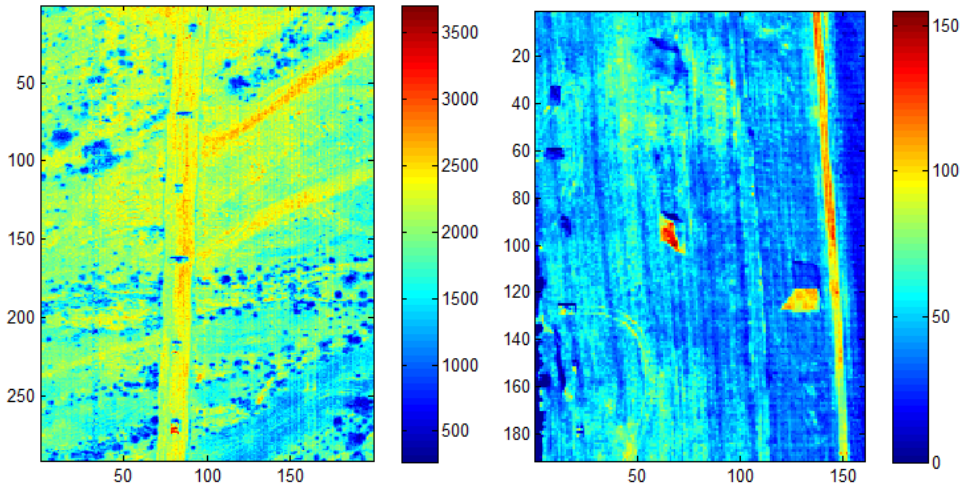


Figure 4.27: HYDICE Threshold Sensitivities.

Based on the HYDICE data, the factor analysis method presented shows great promise. Resulting removals, including zero bands, for all HYDICE images are shown in Table 4.10. Note that cumulative across images, as evidenced by the figures, these bands are very similar to those removed in the literature. However, the factor analysis method provides more flexibility to an specific image when determining what bands to remove. Instead of using a broad range for all images, the method evaluates each image individually. One caveat does need to be mentioned. Some of the HYDICE images contain a few non-noise bands where a sensor artifact or collection error occurred, yielding a line of noise and erroneous pixels in those bands. These bands are not necessarily truly noisy as



(a) ARES1D: Retained. (b) ARES1F: Removed.

Figure 4.28: HYDICE Band Examples: Radiance Values.

they still provide good information for the remaining pixels. However, they can present issues for specific algorithms. The band-selection algorithm here only found some of these for removal. Others such as Smetek [191], Johnson [110], and Jablonski [107] have included some of these bands as well when running their algorithms. More specifics on this occurrence and its impact are included in Chapter 6, but for now those few such bands that were not detected are left in the image. Next, the other HSI data sets are investigated.

4.5.2 AVIRIS.

AVIRIS has 224 bands over 0.4 to 2.5 μm. This range yields peaks in the green wavelengths and diminishes in the higher and lower wavelengths. For the AVIRIS spectrum, deep valleys that go down to near zero occur around 1.4 and 1.9 microns due to water absorbing these wavelengths [2]. Additionally, water absorption also occurs strongly at 2.5 μm [81]. This is obviously the same as the areas of high absorption in HYDICE. Figure 4.29 depicts these wavelengths, where a sub-sample of 1000 random pixels are

Table 4.10: HYDICE Bands ≥ 0.02 Threshold.

Image	Bands Removed	Number Bands Retained
ARES1D	$104:110, 138:153, 157:158, 187:210$	161
ARES1F	$10:11, 103:109, 138:152, 156:157, 180:210$	153
ARES2D	$2:11, 104:108, 139:152, 201, 204, 206, 208:210$	175
ARES2F	$1:8, 104:108, 139:151, 204, 206:210$	178
ARES3F	$103:107, 139:151, 204:210$	185
ARES4F	$1, 103:108, 138:151, 200, 202:210$	179
ARES1C	$104:109, 138:152, 187:210$	165
ARES2C	$103:109, 138:152, 189:210$	166
run03m20	$4:6, 104:108, 139:152, 208:210$	185

shown for each of the four AVIRIS images investigated. The absorption is obvious after Band 100 and Band 150, and it can be seen how bands with low magnitudes (such as 180:200) can at first appear to be noise.

Figure 4.30 shows the correlation of band radiance values to anomalies, as well as the band Bhattacharyya metric. Moderate correlation exists for the Scene1 image, but the correlation magnitudes are still not high. Similar to HYDICE, the Bhattacharyya metric reveals some absorption (for non-zero bands), however, there is no consistency of well-classifying bands across images. Specifically, bands 50-70 for 4Ships2 may be the highest discriminating, while it is not clear if noise or high discrimination is truly responsible for the high values found in Scene1.

Now, using the new factor analysis-based methodology and looking at the number of AVIRIS images that retained a given band when varying the threshold, it was clear that

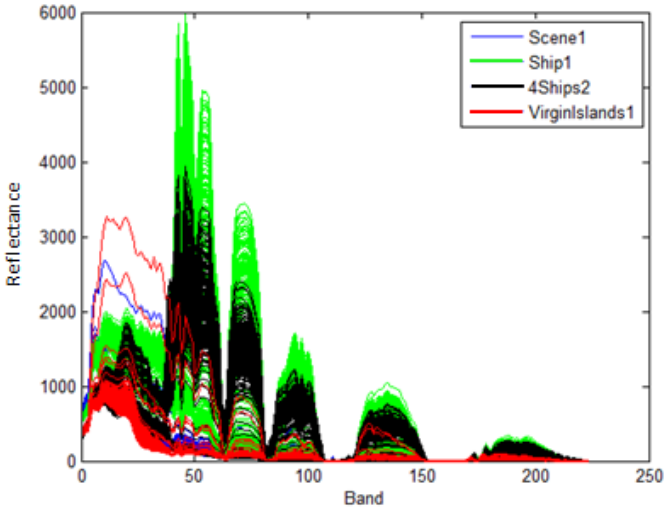
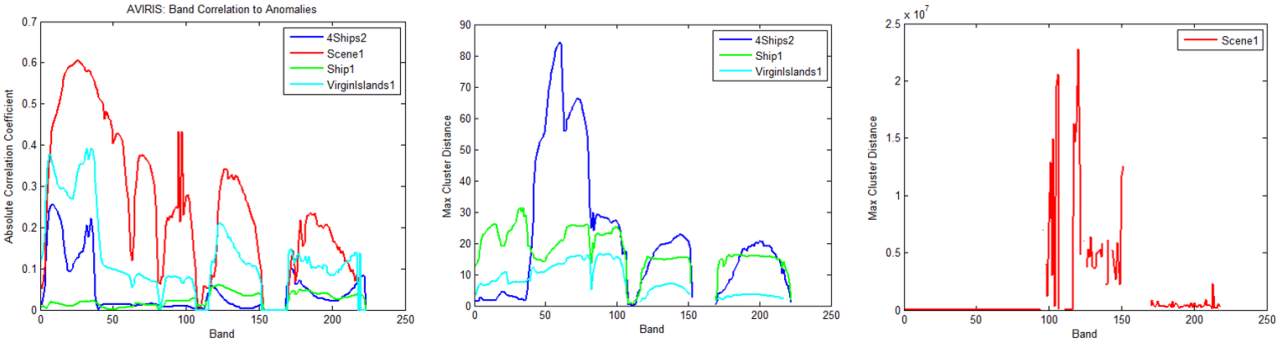


Figure 4.29: AVIRIS Pixel Signatures Sample.

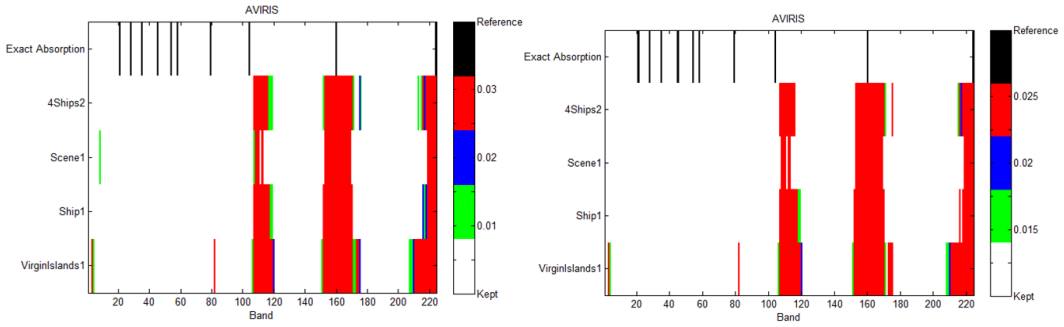


(a) Correlation. (b) $\max_{i,j} B_{i,j,p}$. (c) $\max_{i,j} B_{i,j,p}$.

Figure 4.30: AVIRIS Band Metrics.

again the correct threshold might be $t = 0.02$. Figure 4.31 shows the impact of varying the threshold. Again, all major absorption areas are found, as are potentially noisy bands. To exemplify the importance of removing bands with largely zero pixels before the factor analysis, Band 220 can be considered. This band is a suspected strong absorption band, and

is still removed from every image except Scene1 if largely zero pixel bands are not removed prior. However, Band 220 in Scene1 should be removed, as there are only a few non-zero pixels. These inflate the variance given their magnitude, and become well accounted for in the factor model if kept under consideration.

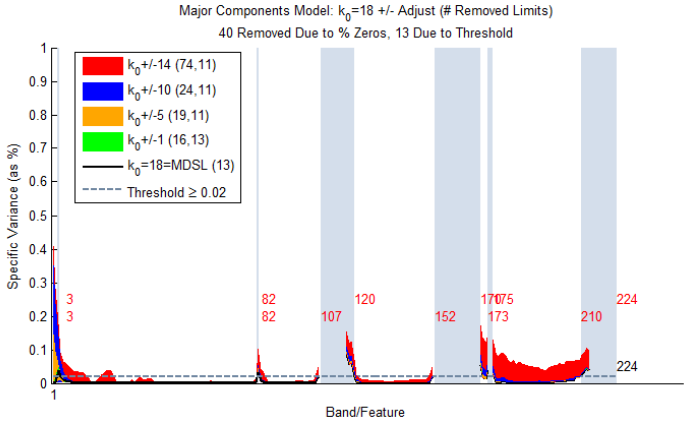


(a) (b)

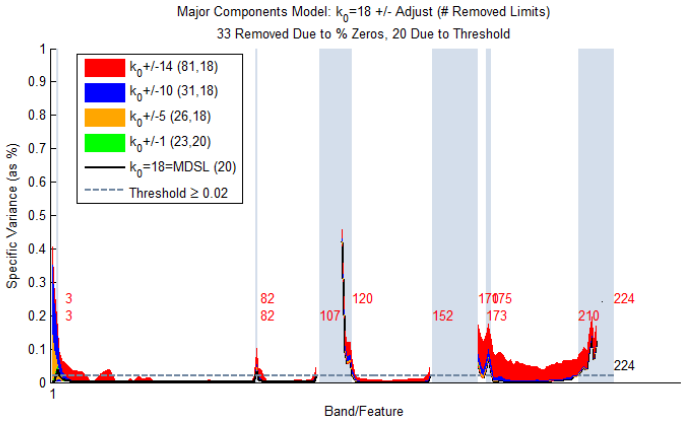
Figure 4.31: AVIRIS Threshold Sensitivities.

To further verify the findings from the HYDICE data, various aspects were again evaluated for the factor analysis algorithm on the AVIRIS data. Figure 4.32 shows both the effect of varying the MDSL cut-off, as well as varying the percent-zero pixel band cut-off for the VirginIslands1 image with a $t = 0.02$ specific variance threshold. This image had the most change in specific variances when varying the MDSL cut-off. As can be seen, these remain fairly consistent until the cut-off is changed significantly. Interestingly, varying the percent-zero pixel cut-off does not affect which bands are ultimately removed (represented by gray fill). This suggests a certain robustness to the method.

It has also been shown that certain, less obvious noisy bands are picked up by the method. Figure 4.33 depicts Bands 81 and 82 for the VirginIslands1 scene. Band 82 has



(a) 0.25% Zero-Pixel Cut-Off.



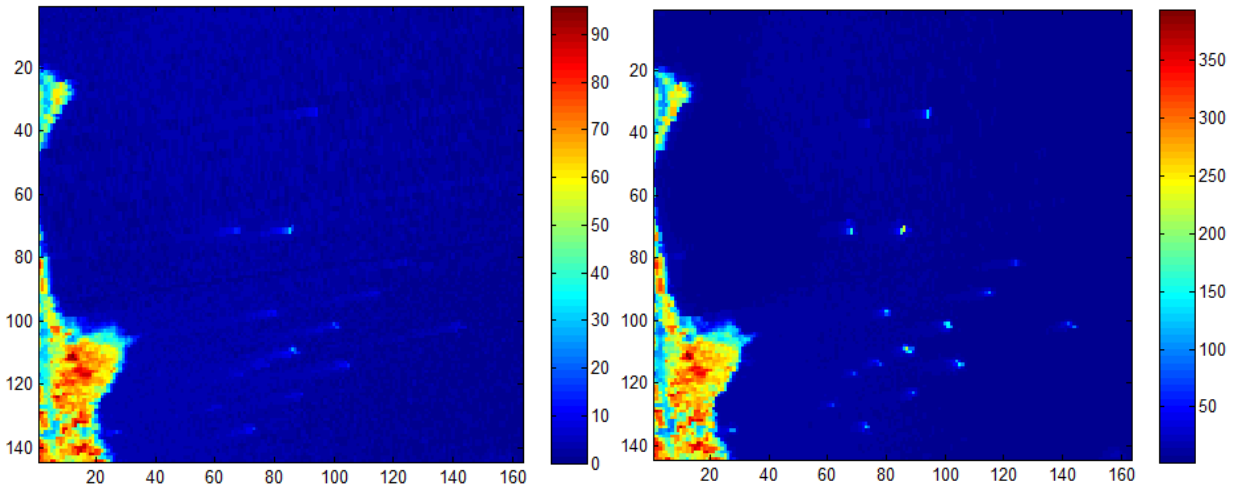
(b) 0.5% Zero-Pixel Cut-Off.

Figure 4.32: AVIRIS Specific Variance Sensitivities.

lower radiance values, and a sort of blurring effect. Table 4.11 shows the bands removed using the $t = 0.02$ threshold.

4.5.3 Pavia.

The Pavia bands lie between 0.43 and 0.86 μm, and thus only very weak absorption occurs [5]. In fact, when varying the specific variance threshold using the factor analysis method, only a very small threshold would remove any bands from consideration. This



(a) Band 82.

(b) Band 81.

Figure 4.33: VirginIslands1 Band Comparison.

Table 4.11: AVIRIS Bands >= 0.02 Threshold.

Image	Bands Removed	Number Bands Retained
Scene1	$108:113, 153:169, 219:224$	196
Ship1	$107:117, 152:170, 216, 218:224$	186
4Ships2	$107:116, 153:170, 175, 216:224$	186
VirginIslands1	$3, 82, 107:120, 152:170, 173:175, 210:224$	171

is shown in Figure 4.34. As this occurs below $t = 0.01$, indicating no presence of noisy bands, none are removed from the Pavia images for this research.

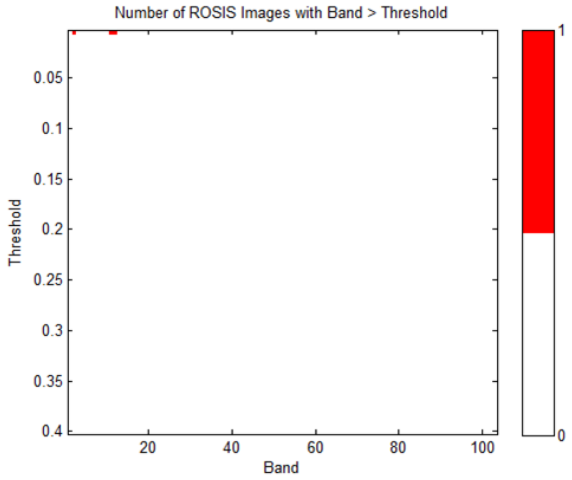


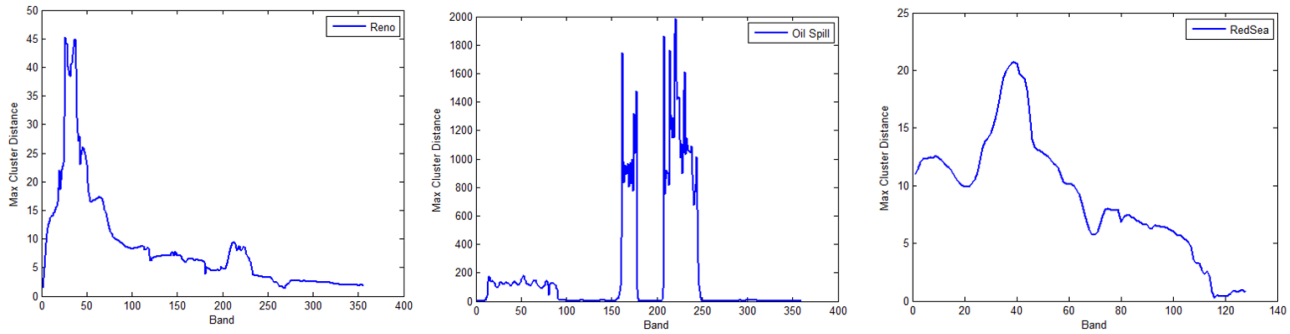
Figure 4.34: ROSIS Images: Threshold Sensitivity.

4.5.4 SpecTIR.

The SpecTIR image collection used in this research consists of three images of varying characteristics, and none have a truth mask. The Reno image covers 0.39-2.45 μm over 356 bands. The Oil Spill image has 360 bands covering 0.39-2.45 μm, collected at 2.2 m ground sample distance. Th Red Sea image, meanwhile, has 128 bands over 0.39-1 μm. Figure 4.35 shows the Bhattacharyya metric for the bands in these images. These indicate general areas of interest in the spectrum that may be desirable to retain.

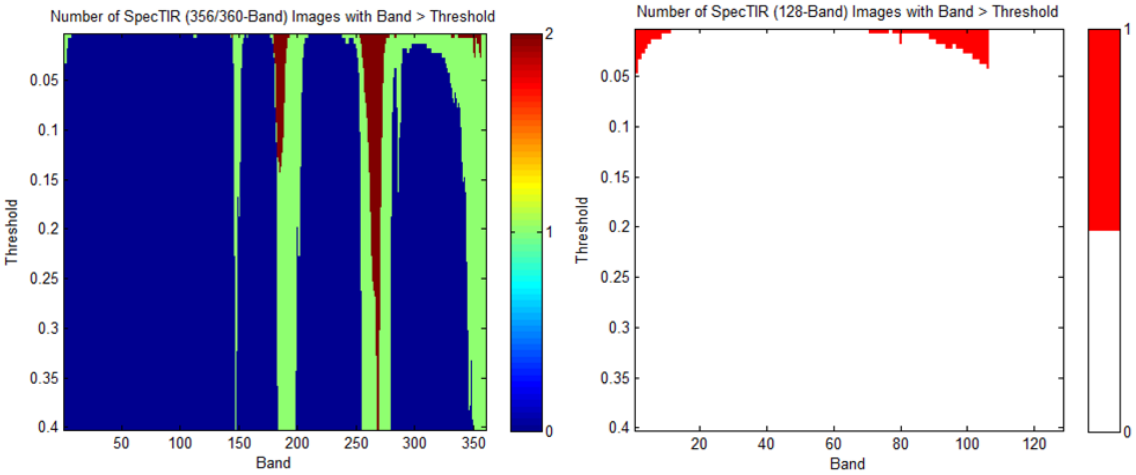
Applying the factor analysis-based method to these images and varying the specific variance threshold yields removals per Figure 4.36. For the images covering 0.39-2.45 μm, the three strong absorption areas are found with a low enough threshold. Additionally, absorption or noisy bands appear present for the Red Sea image. Investigating around $t = 0.02$ further yields those results shown in Figure 4.37.

Lowering the threshold below 0.015 removed all bands in the 300's for the Oil Spill image. Again, $t = 0.02$ appears to be a suitable threshold on the specific variance. Here, all



(a) Reno. (b) Oil Spill. (c) Red Sea.

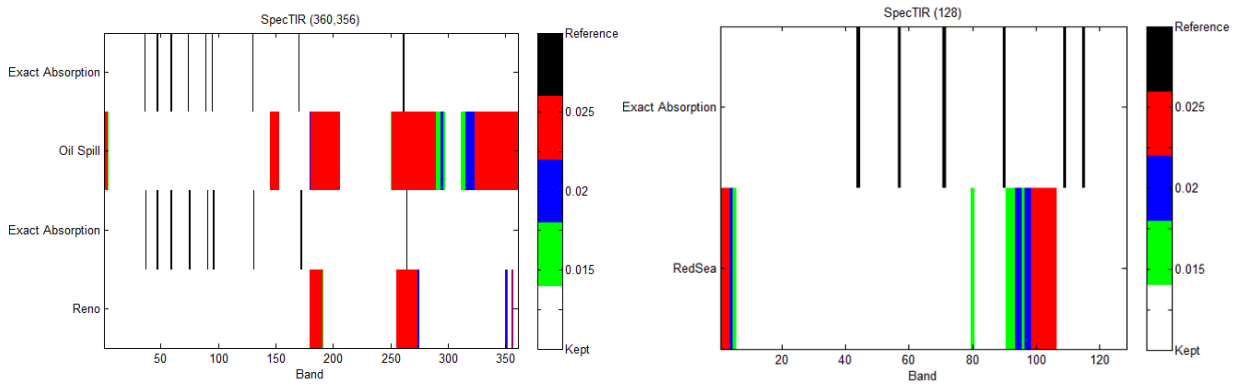
Figure 4.35: SpecTIR Images' $\max_{i,j} B_{i,j,p}$.



(a) Reno & Oil Spill. (b) RedSea.

Figure 4.36: SpecTIR: Threshold Sensitivity.

strong absorption areas are identified, possible moderate absorption is identified in the Oil Spill image, and noisy bands and moderate absorption are identified in the Red Sea image. As an example, consider the bands shown in Figure 4.38 for the Red Sea image. Band 70



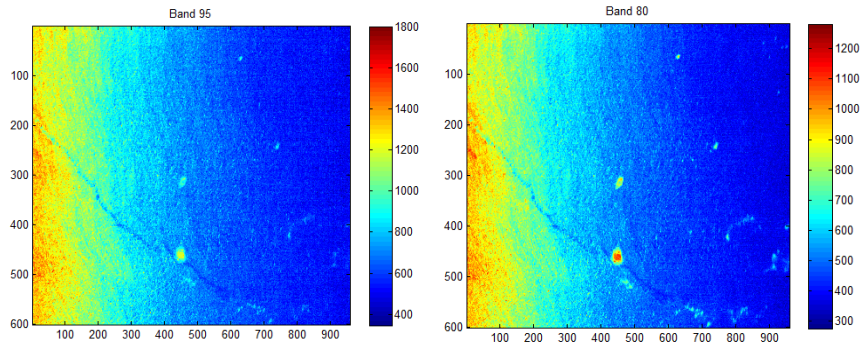
(a) Reno & Oil Spill.

(b) RedSea.

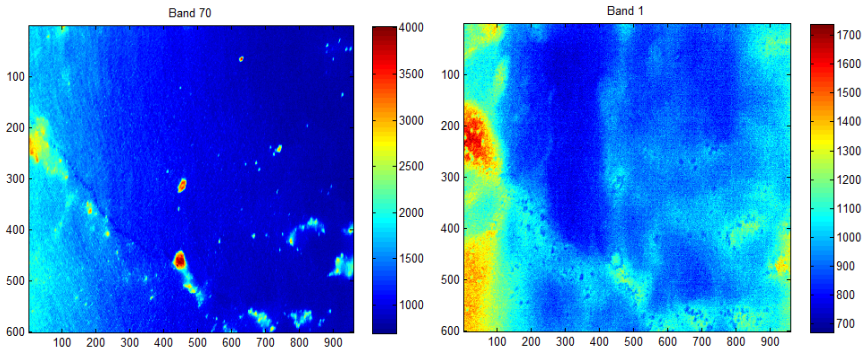
Figure 4.37: SpecTIR: Specific Variance Threshold.

is considered strong over most thresholds, while Band 80 would be considered noisy if the threshold was set to 0.015. In the case of $t = 0.02$, this would not be the case, although band 95 is considered to be noisy. Despite the higher radiance values in band 95 than band 80, the uniqueness of anything within that band is far less pronounced and general noise seems more obvious. Band 1 is a generally noisy band, as can also be seen. The $t = 0.02$ threshold also removes this band.

And so, the $t = 0.02$ threshold and factor analysis-based method developed seems very robust across sensors and image characteristics. The specific bands removed for the SpecTIR images are shown in Table 4.12. Note, none of the bands that had a high Bhattacharyya metric previously are removed. Specifically, bands 155-179 and 206-250 in Oil Spill, bands 10-50 in Reno, and bands 25-50 in Red Sea. This is another indication of the strength of this method and its ability to retain discriminating portions of the spectrum where there is not strong absorption and there is not a large amount of noise.



(a) Band 95. (b) Band 80.



(c) Band 70. (d) Band 1.

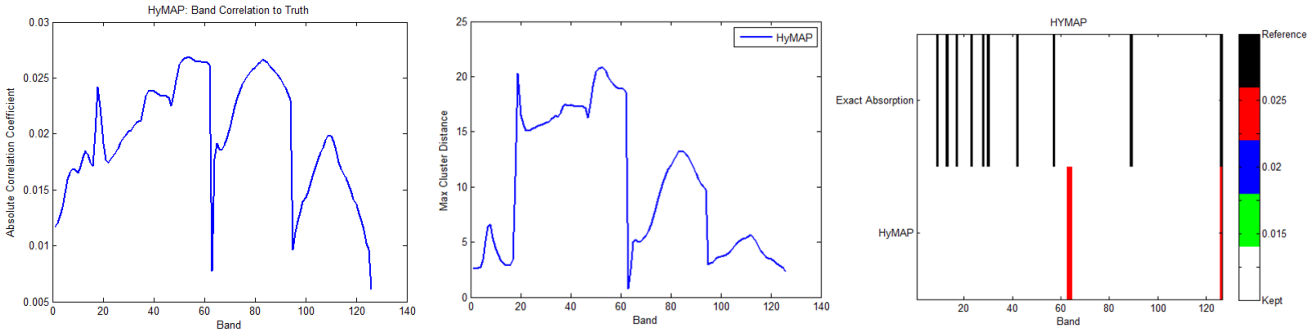
Figure 4.38: RedSea Band Comparison.

Table 4.12: SpecTIR Bands >= 0.02 Threshold.

Image	Bands Removed	Number Bands Retained
Reno	$180:190, 255:274, 350:351, 355:356$	321
Oil Spill	$1:3, 145:152, 180:205, 251:289, 294:295, 315:360$	236
Red Sea	$1:4, 94:95, 97:106$	112

4.5.5 HyMap.

The HyMAP image has 126 bands over 0.453 to 2.496 μm, and Figure 4.39 shows various metrics for this image. The bands have little correlation to non-background pixels, but this is largely due to the small number of such pixels in the image. The Bhattacharyya metric indicated that bands 20-60 may be the best discriminating bands. Using the factor-analysis based method with $t = 0.02$, these bands are retained. In fact, this image showed little sensitivity to varying the threshold, with only three bands removed consistently around the $t = 0.02$ threshold: bands 63, 64, and 126. These correspond to two of the three strong absorption areas in the spectrum, where interestingly and upon visual inspection, the third near 1.9 μm does not manifest in the image. Band 63, to showcase that the method can detect an obvious noise/absorption band that has less than 50% zero pixels, is shown in Figure 4.40. Removal of the three bands leaves 123 retained.



(a) Correlation. (b) Bhattacharyya. (c) Sensitivity.

Figure 4.39: HyMAP Bands.

In conclusion, across images and sensors, a threshold of 0.02 works very well within the factor analysis process. This means that bands with 2% or more specific variance, or equivalently, less than 98% communality in the factor model may be considered noisy or

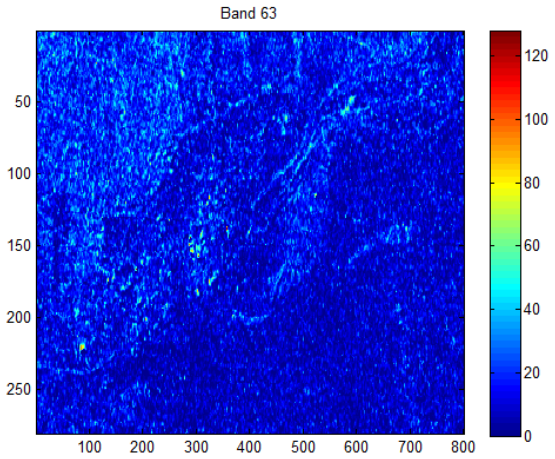
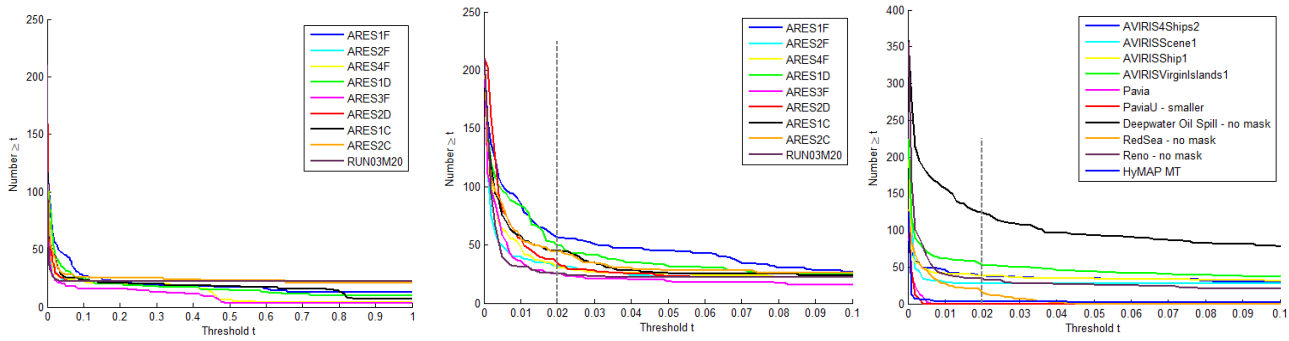


Figure 4.40: HyMAP Band 63.

absorption. This also implies that the factor model is a good model for the HSI image. Lowering the threshold seemed to remove too many bands, while raising it did not remove enough. In seeking another way to automatically find this threshold, the number of bands removed as a function of the threshold was explored. These results are shown in Figure 4.41. Although there is a bend in the curves, somewhat akin to how MDSL is done on PCs, this bend occurs too early on many images such that too many bands might be removed. Once considering that areas of neighboring bands are highly correlated, using a lower threshold corresponding to the bend might leave only the bands needed for a classification task. However, some of those removed would not be truly noisy, perhaps just redundant. As the lower threshold removes entire areas of the spectrum on certain images, this is not desirable for purposes of this research. Rather, any further removal is accomplished equivalently by dimension reduction and other techniques developed and used.

4.5.6 Arcene.

Thus far, only the process from Figure 4.25 has been used on the HSI imagery to remove the absorption and noisy bands. Given that the Arcene dataset has 3,000 false



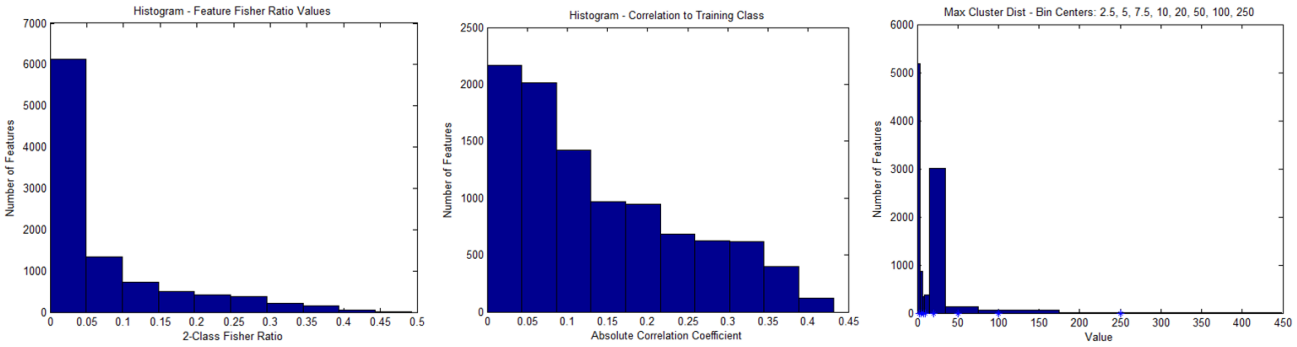
(a) HYDICE. (b) HYDICE Zoom. (c) Non-HYDICE.

Figure 4.41: Number of Bands Removed By Threshold.

features (probes) mixed within its 10,000 features, it may also be good to explore this methodology on that dataset to see just how robust it is to difficult data. With the HSI data, all features are on the same scale, there are far fewer features than exemplars, and features with many zeros are likely among those that should be removed due to radiance generally being positive. The features of the Arcene data are again all on the same scale, but there are far fewer exemplars than features, making it a Small Sample Size (SSS) problem. Additionally, many real features have a large number of zeros. The impact this has on the feature removal process is explored shortly. First, various metrics for the 900-exemplar Arcene dataset are shown in Figure 4.42.

It turns out that these results are fairly similar to what occurs for the 200-exemplar version. As can be seen, very few of the 10,000 total features have a high separation between the two classes, few have even moderate correlation to the truth vector, and few have a high Bhattacharyya metric value. This shows that the real features are not easily found amidst the combined set of probes and real features.

Originally, Arcene was used as a part of the Neural Information processing Systems (NIPS) 2003 contest, where the goal was to identify as few features as possible for good



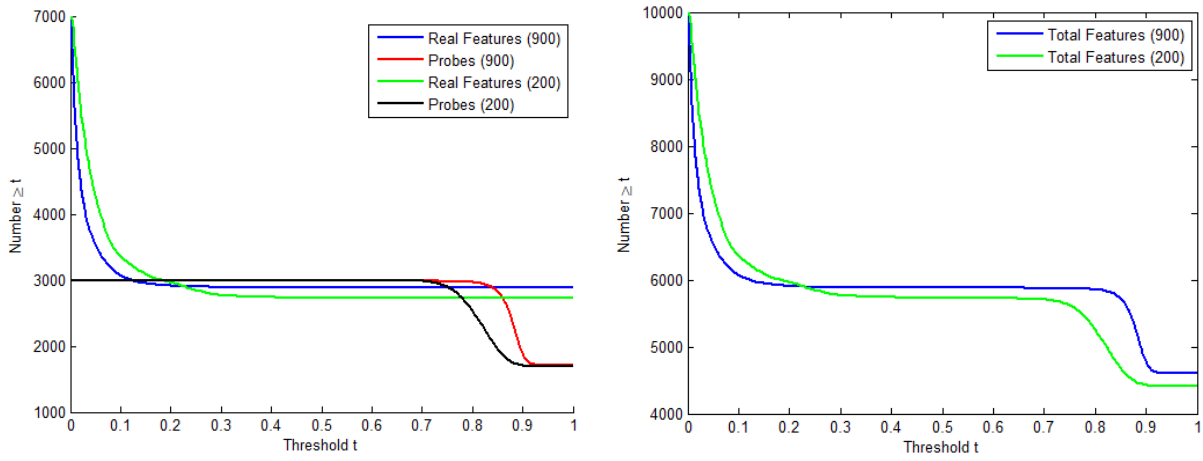
(a) Fisher Ratio. (b) Correlation to Class. (c) Bhattacharyya.

Figure 4.42: Arcene Feature Metrics.

classification on two variants of the Arcene data, as well as other data sets [90]. Lal, Chappelle, and Schölkopf [134] tried to find the best features for classification tasks using, in part, correlation coefficients and Fisher ratios. Their feature subsets contained 47% of the real features, but approximately 6-14% of the probes for Arcene. This is not surprising given the metric values in Figure 4.42. The winning entry found and used 100% of the real features with 30% of the probes in one case, and 11% real and 1% probe in another [134]. Whereas the contest sought a minimal subset of features for classification and also scored based on classification performance, the goal of the factor analysis process here is primarily to find and eliminate the probes, assuming they truly are noisy. Ideally, the process also identifies noisy real features.

With the HSI data, bands with a large percentage of zero values for the radiance were removed in the first step because it is safe to assume that such HSI bands contain noise or collection error, and they can negatively affect the variance in the factor model. In removing these bands, it was shown that it becomes easier to identify remaining bands with large absorption or noise present. For the Arcene data, both the 200 exemplar training/validation set and the 900 exemplar training/test/validation set, this step warrants investigation. 2,723

of the real features have more than 50% zeros, as do 1,692 of the fake features (probes). Further, 535 of the real features have more than 90% zeros, as do 248 of the probes. In fact, 30 real features are entirely zero, as are nine of the probes. Thus, although eliminating the features with a large percent of zero values could still help eliminate probes, it also eliminates some of the real features. For example, Figure 4.43 shows the factor analysis process with varying thresholds t on the 200 and 900-exemplar data sets, including the removal of features with more than 50% zeros.



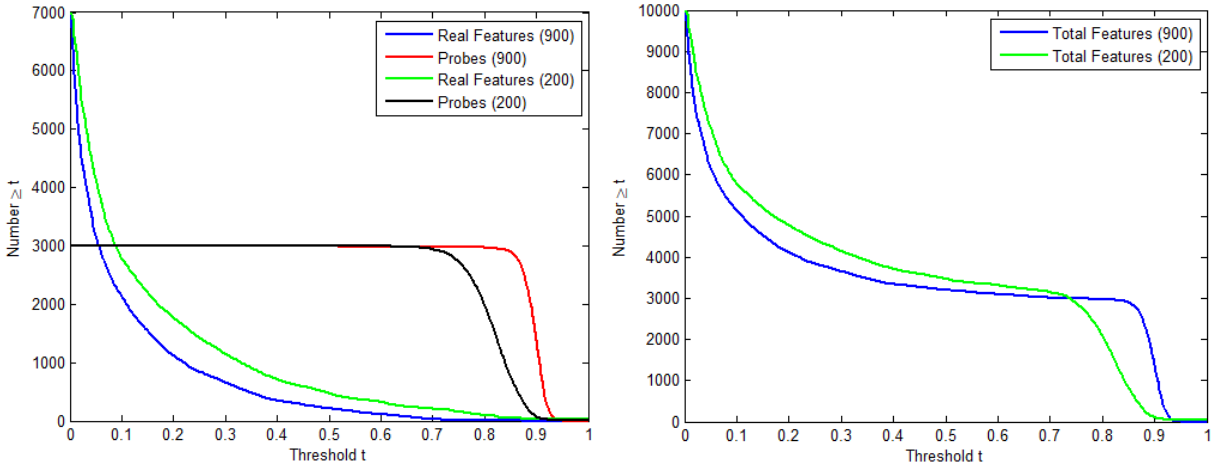
(a) Real vs. Probe.

(b) All Features.

Figure 4.43: Number of Features Removed: Original Process.

A few things are clear from these plots, where it needs to be noted that entire zero and greater than 50% zero exemplar features were assigned a ψ_i of 1 to effectively remove them. All probes are removed until the specific variance threshold goes above 0.7. In addition, no real features, beyond those initially removed, have a specific variance above approximately 0.3. If only the all-zero features are removed in the first step instead, the results change to those shown in Figure 4.44. Again, up to a very high threshold all probes are removed

across data sets. Meanwhile, the threshold has to be higher than before in order to not remove any real features.



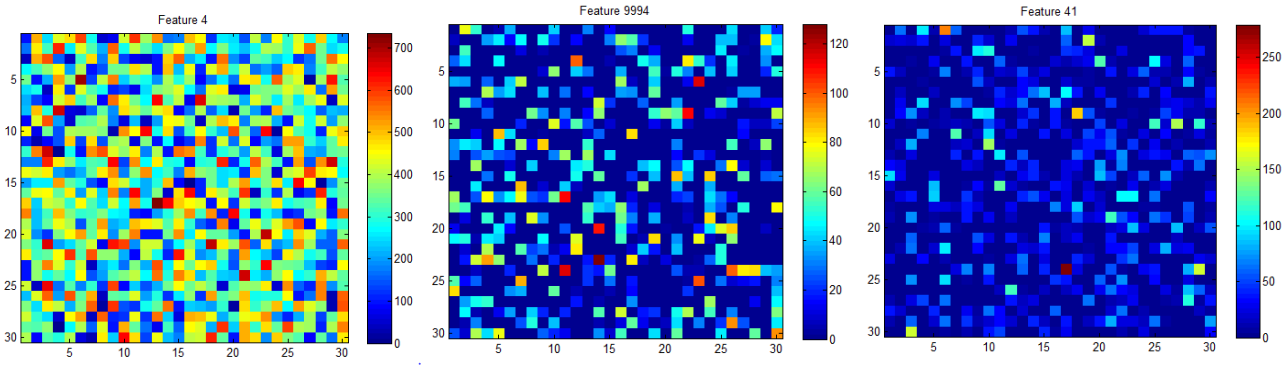
(a) Real vs. Probe.

(b) All Features.

Figure 4.44: Number of Features Removed: Modified Process.

In both cases, the desired threshold appears significantly higher than 0.02, at least at first. Figure 4.45 shows the exemplar values, in a 30×30 grid, for certain features of the 900-exemplar set. Figure 4.45(a)-(c) shows three features with less than 50% zero values and their corresponding specific variances. It seems clear that as the specific variance lowers, there is more information and discrimination present for the exemplars. In contrast, Figure 4.45(d) shows one of the probes with less than 50% zero values. Although the range of values is larger than some of the previous features shown, there also appears to be less discriminating information present for the exemplars. Now, consider Figure 4.45(e)-(f). Feature 9890 has greater than 50% zero values, and appears to have much less information than the previous real features shown. Feature 9895 has less than 50% zero values but

a much higher specific variance than the other real features shown. Again, upon visual inspection it appears this feature has less discriminating information.



(a) Real: $\psi_4 = 0.007$. (b) Real: $\psi_{9994} = 0.046$. (c) Real: $\psi_{41} = 0.131$.

(d) Probe: $\psi_{39} = 0.912$. (e) Real: $\psi_{9890} = 0.055$. (f) Real: $\psi_{9895} = 0.31$.

Figure 4.45: Feature Examples.

This analysis implies that the factor process is, in fact, robust to data such as Arcene. Here, convergence for the varimax rotation can require many more iterations due to the number of features. As it is unnecessary unless interpretation of the factors is a goal, this step of the process should probably be removed for large data sets in practice. It is difficult to quantify exactly which of the real features are good for classification, but it is clear that no matter how the first step of the process is treated, a specific variance threshold of 0.3

would remove all probes and relatively few real features. Arguably, an even lower threshold would also remove noisy or low information real features based on looking at the features. It is not entirely clear as to whether all real features with more than 50% zero values should be removed as a first step, but there is some evidence to support no change to the process in this regard. This also makes sense intuitively. However, it is important to mention that a threshold higher than 0.02 does need to be used on this data, vice HSI, because it would remove too many features. Fortunately, this can be explained.

Recall, Arcene is a SSS problem. When doing the MDSL cut-off before creating the factors and rotation, only 86 components are kept for the 900-exemplar data, and only 36 are kept for the 200-exemplar data. Although this maintains a large percentage of the variance found in the data in both cases, thousands of features are being condensed into tens of factors for only hundreds of exemplars. This is very different than in HSI. Whereas if $N > p$, p factors can be estimated, here the process is limited by N. In the case of the original process, *i.e.*, when removing all features with more than 50% zeros values, $t = 0.02$ would leave less than 2200 of the real features. Although these may be the 'best' features, $t = 0.1$ or slightly higher could be a better choice for the threshold as it would leave most of the remaining real features while also removing all probes, as evidenced in Figure 4.43. As a result, the process shows great promise in identifying noisy features, or features that are unlikely to help discriminate between classes.

V. n-Dimensional Visualization

5.1 Literature Review

With large-dimensional data sets, visualization of the data, its structure, and any discrimination caused by methods becomes very difficult. Dimension reduction techniques allow for the plotting of a smaller number of dimensions, but without simple data, often more than three dimensions are still required for any intended purpose. Although PCA and other methods optimize based on data characteristics, they are not guaranteed to yield a meaningful representation of class information in two or three dimensions. This visualization problem has been approached several ways in the literature, none without limitation. Some become computationally expensive as the number of data features increases, others are not very intuitive, and more do not lend themselves to visualizing greater than a moderate number of features. Surveys of various methods include those by Chan [44], Kehrer and Hauser [118], Keim [119], Kromesch and Juhasz [129], and Grinstein, Trutschl, and Cvek [82].

The high-dimensional data visualization problem has been approached both from the standpoint of multivariate data, as well as for visualization of Pareto fronts in multi-objective optimization. A few techniques from both of these areas are leveraged here. Originally, the visualization research in this chapter was formulated with the purpose of being able to visualize the decision boundaries for high-dimensional data and non-linear methods. However, as the research progressed, it became apparent that the techniques were more generalizable and useful for multivariate data as a whole. For example, one purpose is to use the visualization to identify general characteristics of a multivariate dataset such as class overlap and outliers. In the application of classification, a purpose is to enable data complexity comparisons, possible class identification, and an evaluation of linear or non-linear algorithm appropriateness.

In order to develop an improved n-dimensional visualization, it is necessary to first review existing visualizations and their limitations. Perhaps the simplest method is to use a '3-at-a-time' approach in order to generate $\binom{p}{3}$ plots, one for each set of three features, for investigation of a total set of p dimensions or components. A full set of scatterplots is a common method, where each feature is plotted against other feature [129]. Both of these techniques are problematic in that it can be very difficult to evaluate total class and structure differences across the many plots. Another simple method is referred to as Parallel Coordinates, where each feature is normalized according to its range and plotted on a tick of the x-axis, connecting exemplars' feature values with lines [105]. This is shown in Figure 5.1 for the Fisher Iris data. Parallel Coordinates has clear limitations as the number of exemplars and features increases [169]. Variants of parallel coordinates also exist, such

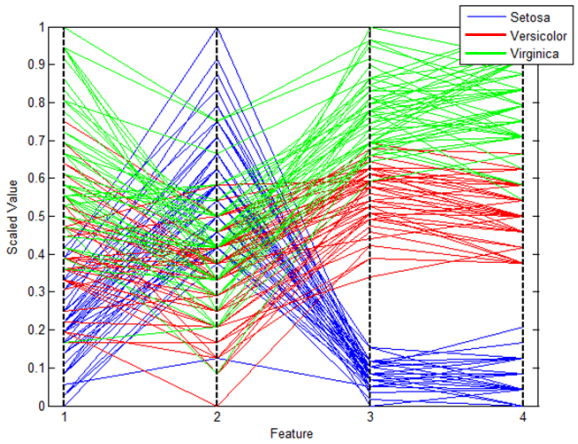


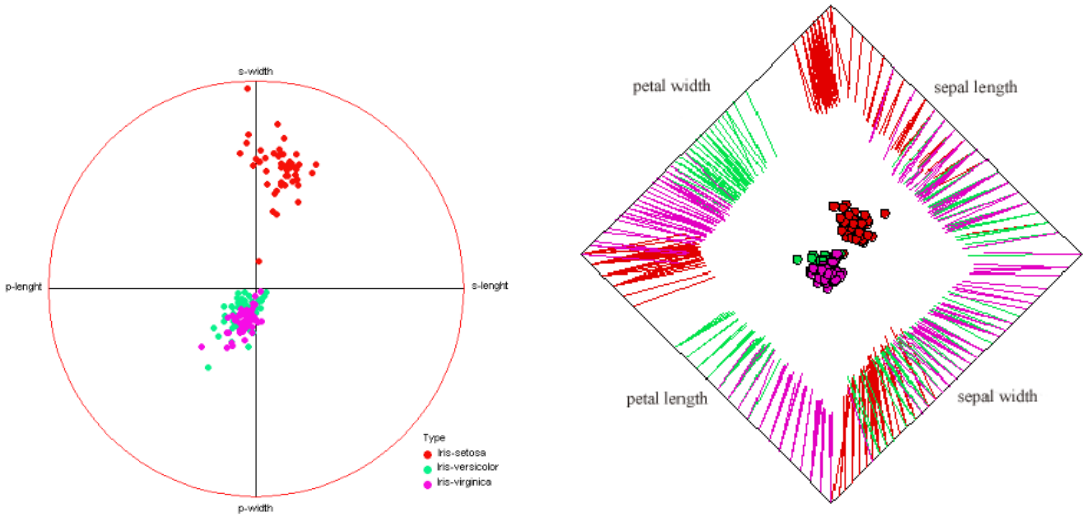
Figure 5.1: Parallel Coordinates Example.

as connecting normalized feature values radiating from the center of a circle akin to a radar graph [129], or using parallel dual plots as presented by Xu et al. [222]. These results can still be difficult to interpret as the number of features grows due to overlapping lines and points.

Several other methods have been developed. These include: iconographic (or glyph) displays, multi-line graphs, by-feature heat maps, polar charts (a circular form of parallel coordinates), logic diagrams of the features, survey plots of the features, and hierarchal methods [44, 82]. Additional methods include mosaic matrices [126], using a hyperbox [13], multiple frames and non-linear magnification [117], and table lens where a line has length based on an attribute and a color based on another [174]. All of these methods have interpretation and clutter issues as the number of features grows. Principal curves is a method that uses a smooth 1-D average of the data points allowing for a non-linear representation, but it does not necessarily provide advantage over using self-organizing maps to provide a better non-linear reduction and it does not even fully separate a dataset such as Fisher Iris [48, 95].

RadViz is a technique that places dimensional anchors (the features) around a circle, with spring constants utilized to represent relational values among points [98]. Each spring is attached to an anchor and a data point, where the data is then displayed where the sum of all spring forces is zero. PolyViz is a similar construct, with each feature anchored instead as a line [82]. These latter visualizations are shown in Figure 5.2 for the Fisher Iris dataset. Again, these become less useful as the number of features and exemplars grow.

Yet another method, developed for Pareto front visualization is Hyperspace Diagonal Counting and the Hyperspace Pareto Frontier. This method, developed by Agrawal, Lewis, and Bloebaum [11] is based on the premise of Cantor's counting method. Exemplars are mapped to a line by counting along hyperdiagonal bins that move away from the origin, as shown for two features in Figure 5.3(a). Features are split into two groups, one for each axis, before each group is counted independently. The bins and their counts, moving away from the origin on these diagonals are then plotted. This representation thus can also give a density along the respective hyperdiagonals. A sample is shown in Figure 5.3(b).

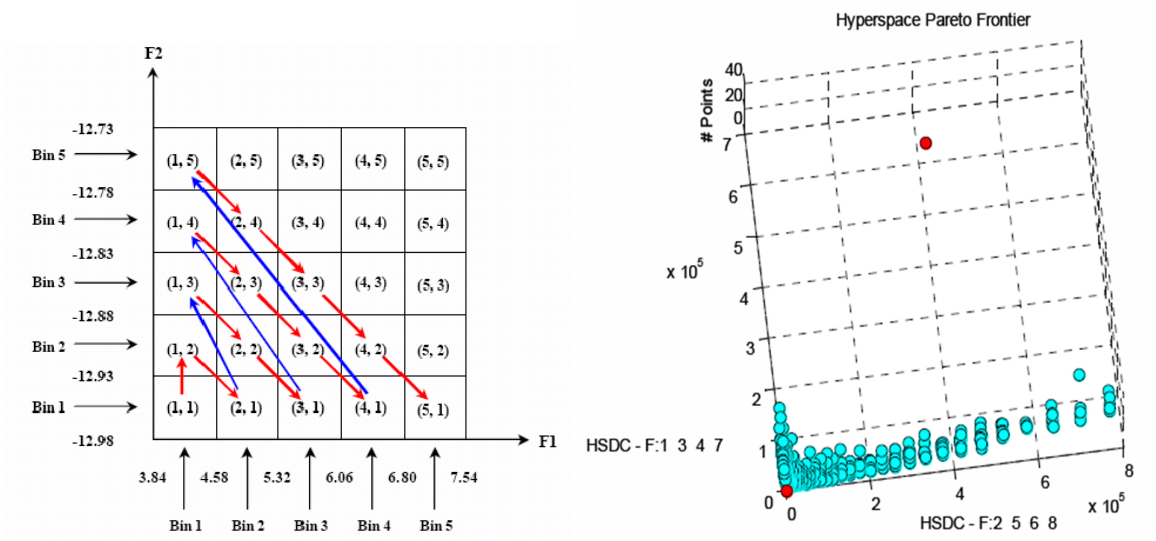


(a) RadViz. (b) PolyViz.

Figure 5.2: Anchor-Based Visualizations [82].

Unfortunately, this method is not necessarily efficient, is variable depending on the groups chosen, and sometimes gravitates values towards the axes [169].

As mentioned, any focus on visualizations in this research serves two purposes: to provide a representation that can depict decision boundaries or changes in these boundaries in a meaningful and comparable way, and to evaluate the complexity of a dataset. Ideally, the visualization is also intuitive so that results can be interpreted easily. Bertini, Tatu, and Keim [27] suggested the evaluation of high-dimensional data visualizations via: 1) the extent to which data groupings are maintained (clustering), 2) the extent to which systematic changes in one dimension are accompanied by changes in the others (correlation), 3) the maintenance of outliers, 4) the clutter of the visualization, 5) feature preservation, and 7) complex pattern (a literal catch-all). In Section 4.2, a useful visualization that satisfies (3) and (4) to evaluate dataset complexity was discussed and



(a) Diagonal Counting. (b) Visualization.

Figure 5.3: Hyperspace Diagonal Counting [11, 169].

applied. In Section 5.3, a visualization that could potentially satisfy all of the above properties is further developed and applied.

5.2 Dimensionality Reduction and Random Projections

Before proceeding to development of the visualization, it is important to discuss dimension reduction. Clearly, if the data can be reduced in dimensionality then the lower-dimensional data would be easier to visualize. However, using dimensionality reduction in this manner, as an input to the visualization, can cause two issues. First, properties of the original data need to be maintained in the lower-dimensional representation for the visualization to have meaning. Second, the reduced data needs to be interpretable such that the resulting coordinates have a meaning to the user.

Johnson and Lindenstrauss [112] showed that any set of N points in high-dimensional Euclidean space can be mapped into an $O\left(\log N/\epsilon^2\right)$-dimensional Euclidean space such

that all pairwise distances between points are preserved to within a factor of $(1 \pm \epsilon)$. Such a mapping would prove powerful when generating a visualization for high-dimensional data. Dasgupta and Gupta [62] later proved a related result shown here as Theorem 5.2.1.

Theorem 5.2.1. *[62] For any $0 < \epsilon < 1$ and $N \in \mathbb{N}$, let k be a positive integer such that $k \geq 4\left(\epsilon^2/2 - \epsilon^3/3\right)^{-1} \ln N$. Then for any set V of N points in $\mathbb{R}^p$, there is a map $f : \mathbb{R}^p \to \mathbb{R}^k$ such that for all $u, v \in V$, $(1-\epsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2$.*

Furthermore, this map can be found in randomized polynomial time.

Unfortunately, no way to explicitly and consistently generate this mapping has been determined. There have been several random methods developed, however, to find projections that satisfy the Johnson-Lindenstrauss Lemma and related results probabilistically. An empirical study of several of these methods was conducted by Venkatasubramanian and Wang [207], where they found distance errors to be distributed exactly as predicted by the lemma. They also found a target dimension of $\lceil 2 \ln N/\epsilon^2 \rceil$ sufficient for a desired error rate. These findings already point to issues with such methods, as target dimensionality is quite large even for moderate N and ϵ. Further, the projections are random in nature, meaning that to achieve the desired error many projections may need to be formed. As Fern and Brodley noted [72], random projection is highly unstable. One specific instance of such methods is the Fast Johnson-Lindenstrauss Transform, where a randomized Fast Fourier Transform is followed by a sparse projection [10]. Blum [32] showed that random projection could maintain linear separability, assuming the original data had a large separation margin.

Achlioptas [8, 9] developed a simple and efficient projection that satisfies the Lemma, shown in Theorem 5.2.2, where the Lemma holds probabilistically as a function of the number of sample points and a parameter β.

Theorem 5.2.2. *[9] Let V be an arbitrary set of N points in $\mathbb{R}^p$, represented as a $N \times p$ matrix X. Given $\epsilon, \beta > 0$ let $k_0 = \dfrac{4+2\beta}{\epsilon^2/2 - \epsilon^3/3} \ln N$. For an integer $k \geq k_0$, let R be a $p \times k$*

random matrix with $R(i,j) = r_{ij}$ *where* $\left\{r_{ij}\right\}$ *are independent random variables from either one of the following two probability distributions:*

$$r_{ij} = \begin{cases} +1 & \textit{with probability } 1/2, \\ -1 & \textit{with probability } 1/2, \end{cases}$$

$$r_{ij} = \begin{cases} +\sqrt{3} & \textit{with probability } 1/6, \\ 0 & \textit{with probability } 2/3, \\ -\sqrt{3} & \textit{with probability } 1/6. \end{cases}$$

Let $E = \dfrac{1}{\sqrt{k}} XR$ *and let* $f : \mathbb{R}^p \to \mathbb{R}^k$ *map the i-th row of X to the i-th row of E. With probability at least* $1 - N^{-\beta}$*, for all* $u, v \in V$, $(1-\epsilon)\,\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\epsilon)\,\|u-v\|^2$.

Li et al. [140] adjusted the probability distributions for the Achlioptas projections to yield a more efficient sparse projection that still maintained local distances in the expectation. Some algebraic manipulation of the Achlioptas result enables a more direct evaluation of such a random projection, and leads to the new Corollary 5.2.2.1. This enables a lower bound on k as a function of N, ϵ, and the probability that distance is maintained q.

Corollary 5.2.2.1. *For any* $0 < \epsilon < 1$*, any set of N points V in* $\mathbb{R}^p$*, and* $p \geq k \geq 4\left(\epsilon^2/2 - \epsilon^3/3\right)^{-1} \ln N$*, there exists a map* $f : \mathbb{R}^p \to \mathbb{R}^k$ *that can be found in randomized polynomial time such that for all* $u, v \in X \subset \mathbb{R}^p$, $(1-\epsilon)\,\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\epsilon)\,\|u-v\|^2$. *Specifically, the projection from Achlioptas is such a mapping for* $k \geq \left(4 - \dfrac{2\ln(1-q)}{\ln N}\right)\left(\epsilon^2/2 - \epsilon^3/3\right)^{-1} \ln N$*, where q is the lower bound on the probability that the distance between any two points in V is maintained to within a factor of* $(1 \pm \epsilon)$.

Proof The first part of the Corollary is simply a restatement of Theorem 5.2.1.

For the result on Achlioptas' projection, recall from Theorem 5.2.2 that $q \geq 1 - N^{-\beta}$.

$\Rightarrow N^{-\beta} \geq 1 - q \Rightarrow -\beta \ln N \geq \ln(1-q) \Rightarrow \beta \leq -\dfrac{\ln(1-q)}{\ln N}$.

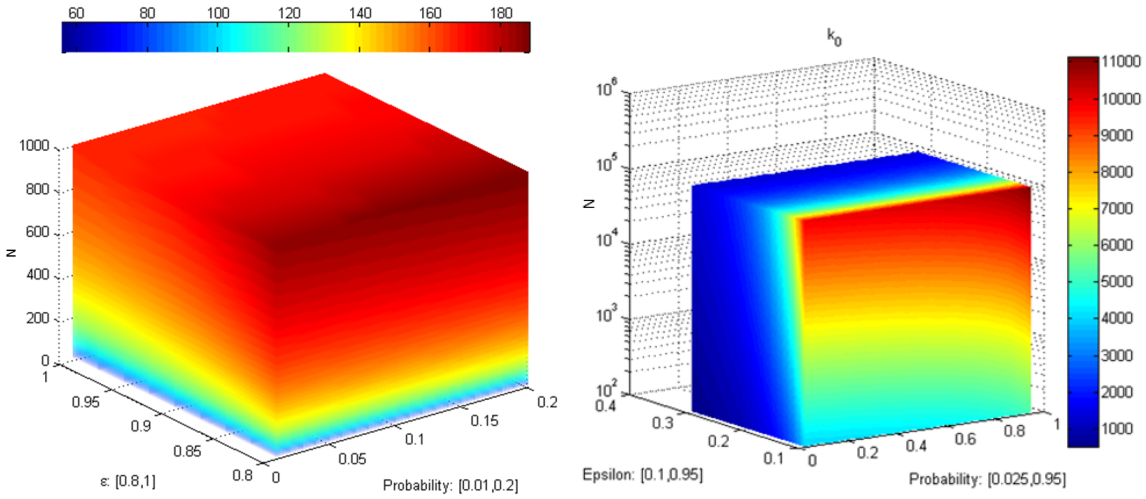
This also implies that β is non-negative, which maintains that $q \in [0, 1]$.

Now, this new result for β can be applied to the bound for k.

From Theorem 5.2.2 it follows that $k \geq k_0 = \dfrac{4 + 2\beta}{\epsilon^2/2 - \epsilon^3/3} \ln N$.

This yields that $k \geq k_0 \geq \left(4 - \dfrac{2\ln(1-q)}{\ln N}\right)\left(\epsilon^2/2 - \epsilon^3/3\right)^{-1} \ln N$.□

These findings are important to discuss, because although they suggest the existence of projections that maintain local information for all points, in practice such methods often exhibit high error. For example, Figure 5.4 shows the bound, k_0, from Corollary 5.2.2.1 over different ranges of ϵ, N, and q. Even with a relatively small number of points, the required dimensionality for the reduction is very high, and this holds with low probability and high error in the maintenance of local distances. Considering the case of $N = 200$, $\epsilon = 0.9$ and $q = 0.1$, $k_0 \geq 132$; and, in fact, the bound increases with larger N and q and as ϵ decreases.



(a) (b)

Figure 5.4: k_0 Values as a Function of ϵ, q, and N.

Whereas Johnson-Lindenstrauss focuses on local information, methods such as PCA can yield a global guarantee on error, *i.e.*, reconstruction error. Although this can be useful in certain applications, for data visualization this is problematic in that the relationships

between points may change. Additionally, direct interpretation of the components, the linear combinations of features, can be problematic. Multi-dimensional scaling is another technique that tries to approximate the Johnson-Lindenstrauss mapping, but when using L_2 for similarity, the embedding is the same as PCA scores and does not guarantee the maintenance of local information [61]. van der Maaten and Hinton [150] developed the t-Stochastic Neighborhood Embedding (t-SNE) algorithm in an effort to maintain the relational structure of data when embedding in a lower-dimensional space. They used t-SNE to generate a relatively successful cluster visualization of a 6,000 exemplar subset of the MNIST dataset. However, t-SNE models Kullback-Liebler divergence between neighborhood conditional probabilities for all exemplars in the original and transformed spaces. Such an approach is computationally expensive, as the conditionals are computed for all exemplars and the transformed space is updated iteratively via a gradient approach. Further, feature information is lost and only a measure of aggregate proximity is maintained. The algorithm also attempts to mitigate crowding of points, thus artificially adjusting the closeness of certain exemplars and clusters in the visualization. Nonetheless, t-SNE and projections are used for comparison, on occasion, to the visualization developed here. The t-SNE code used to generate those visualizations was taken directly from van der Maaten and Hinton [148].

5.3 Hyper-Radial Visualization and Improvements

Chiu and Bloebaum [56] developed Hyper-Radial Visualization (HRV) in order to yield a straightforward method that would not suffer any of the problems of other n-dimensional visualizations for Pareto fronts when evaluating solutions to multi-objective optimization problems. Their intent for the visualization was to be able to compare design solutions to one another and desirability relative to a utopia, or ideal, solution. Here, as HRV is improved, the focus for the visualization becomes more about feature groupings and

identifying data structure. First, HRV is introduced within the multi-objective optimization problem context. Then, it is expanded to multivariate data.

Letting F_i denote the i-th objective function in the multi-objective problem, Chiu and Bloebaum [56] began by normalizing these objective function values for p objectives using,

$$\widetilde{F}_i = \frac{F_i - F_{i,min}}{F_{i,max} - F_{i,min}} \in [0, 1] \tag{5.1}$$

for $i = 1, \ldots, p$, where $F_{i,min}$ and $F_{i,max}$ were the minimum and maximum values of the set of solutions in that objective. This normalization changes the scale of features relative to one another, but maintains information found within each feature. It also later ensures coordinates in a [0, 1] interval for the visualization, helping to prevent outliers from greatly skewing the visualization.

Next, they grouped objectives into two sets,

$$G_1 = \left\{\widetilde{F}_1, \widetilde{F}_2, \ldots, \widetilde{F}_s\right\} \tag{5.2}$$

and

$$G_2 = \left\{\widetilde{F}_{s+1}, \widetilde{F}_{s+2}, \ldots, \widetilde{F}_p\right\}, \tag{5.3}$$

where $s = \lceil p/2 \rceil$. These groups were typically not chosen in any special way other than by a current order of objectives. For each group, a Hyper-Radial Calculation (HRC) value was computed for each solution as,

$$HRC_j = \sqrt{\frac{\sum\limits_{i \in G_j} \widetilde{F}_i^2}{n_j}}, \tag{5.4}$$

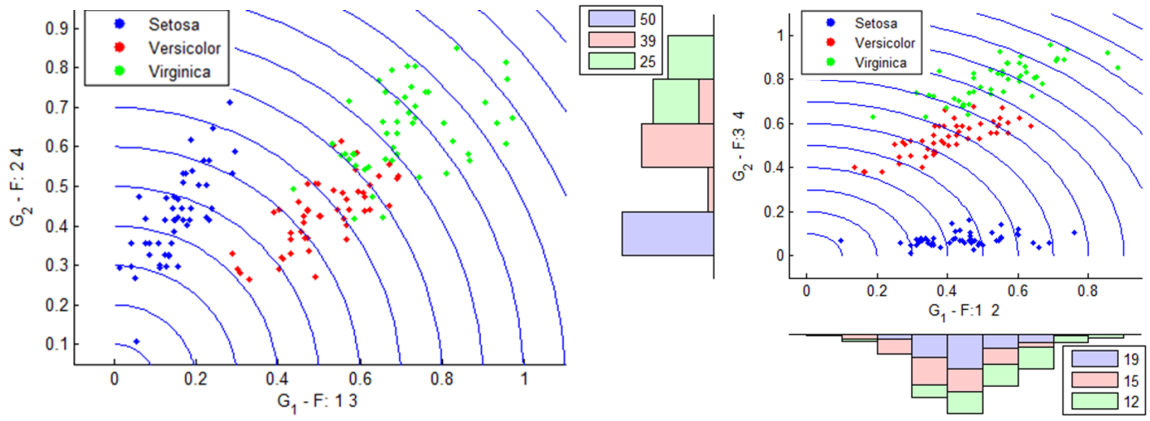
where $j = 1$ or 2 for G_j, and n_j is the number of objectives in the group j. For an unbiased representation, the two groups were kept equal in size. This meant that for an odd number of objectives, one group was given a dummy objective consisting entirely of zeros. Additionally, they used a third overall Hyper-Radial value,

$$R = HRC_1^2 + HRC_2^2. \tag{5.5}$$

This represented the squared radius of a solution from the minimum reference point (the overall minimum objective values in the set). The *HRC* and *R* values then constituted metrics that helped determine the quality of a solution relative to an ideal solution. Curves of constant distance were also added to the HRV visualization to enable color-coding of distance regions from the ideal solution.

The HRV has great promise, in that it is easily interpretable and efficient to generate. The coordinates themselves are simple weighted Euclidean distances from the minima for each group of normalized objectives. This representation maintains relative geometry of the data without a true transformation taking place. Further, minima occur at zero and maxima occur at one on the coordinates, making it easy to relate positioning of solutions to one another. This simplicity inspired an investigative look into possible improvements to HRV for purposes of visualizing multivariate data. Of course, it is clear that the HRV algorithm can be applied directly to a $N \times p$ dataset X, where the features take the place of objectives. The exemplars of Fisher Iris, using HRV without the R value and treating each feature as an objective and each exemplar as a solution, are depicted in Figure 5.5(a). Here, the axes labels denote the group number followed by the feature numbers within that respective group.

In order to improve the visualization, its limitations first needed to be understood. One limitation is HRV's reliance on appropriate groups for the *HRC* values. A way to overcome this was to identify a criterion for optimal groups. This is discussed in Section 5.3.1. Another limitation is that different data can map to the same point and that many points can map to the same radial, *e.g.*, Figure 5.6, although the latter is not necessarily as much of an issue. It is true for any visualization/mapping that only so much information can be displayed in two or three dimensions. Thus, the possibility of mapping to the same point is generally unavoidable no matter the approach. However, in order to mitigate this issue somewhat, stacked class histograms are added to the HRV visualization axes. The



(a) Old. (b) Improved.

Figure 5.5: HRV: Fisher Iris.

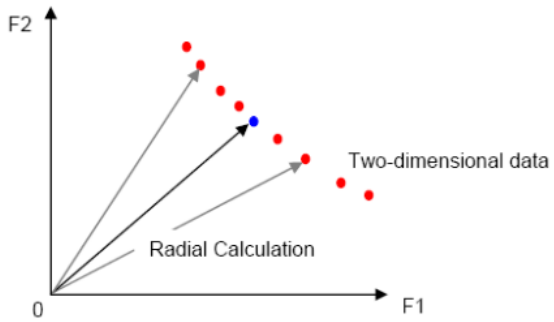


Figure 5.6: HRV Radial [164].

maximum number of class exemplars over all bins is added as a legend to the histogram to give a notion of scale. The bin widths are chosen by Scott's rule [185]. This, coupled with the optimal groupings helps to avoid unnecessary crowding of exemplars in the visualization for the purposes here. t-SNE tried to mitigate this crowding issue within its approach, but it is shown that the method developed here still provides a better visualization on many data sets. Fisher Iris, with optimal HRV groupings and improvements, is depicted

in Figure 5.5(b). This visualization is deemed optimal in that the three classes have clear class boundaries, with the Setosa class being obviously different than the others even in an unsupervised setting.

The importance of the Euclidean metric to the visualization was also assessed. Consider again the Fisher Iris data, where it is normalized and split into two groups as before. However, instead using the Mahalanobis distance $(\widetilde{x}-\widetilde{\mu})^T\,\widetilde{C}^{-1}\,(\widetilde{x}-\widetilde{\mu})$ with $\widetilde{C}$ and $\widetilde{\mu}$ denoting the covariance matrix and mean vector for the normalized data $\widetilde{X}$, respectively, Figure 5.7 is the generated visualization. Note, the distance was scaled to $[0, 1]$ after computation for direct comparison. Because the covariance weights the distance values, true outliers become more pronounced. This ends up condensing portions of the visualization, and makes it most effective at outlier identification. Many other distance metrics, inclduing several of those from Section 4.1 were investigated, but the resulting visualizations demonstrated a high correlation to the L_2-based results, or they did not decompose as cleanly and/or spread the data as well as the L_2 norm on a common $[0, 1]$ scale.

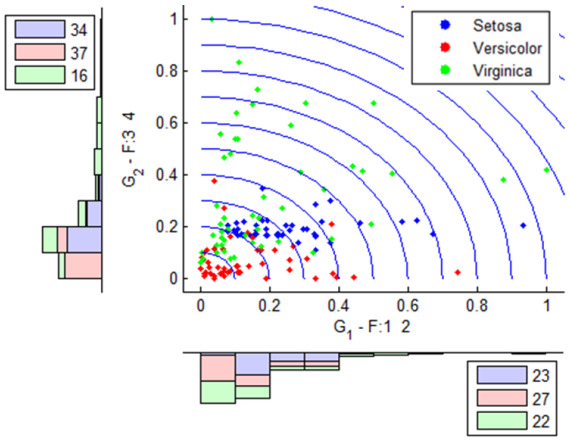


Figure 5.7: HRV Using Mahalanobis Distance.

5.3.1 Determining Optimal Groupings.

For purposes of comparison and evaluation in this section and following sections, the data sets from Table 5.1 are used primarily. The ratio shown to quantify complexity is that of Equation 3.82, and all zero features are removed as they provide no value to a visualization.

Table 5.1: Datasets Extended Fisher Ratios.

Name	Ratio ($/p$)
Fisher Iris	30.78 (7.70)
Vertebral Column (3 Class)	2.98 (0.50)
Breast Cancer (Diagnostic)	10.93 (1.21)
Wine Quality	3.45 (0.31)
Wine	13.93 (1.07)
MNIST	106.60 (0.15)

In order to find criteria for optimal groupings, as well as a methodology to determine these optimal feature groupings for the visualization, it is important to determine a formulation of the problem. Assuming some objective that can be optimized as a function of the coordinates, each feature can be placed in one of two groups via a binary variable. This formulation is shown in Equation 5.6, where J_t is the objective indexed by t so different objectives can be used, x_i is the value of the exemplar x in the i-th feature, $\widetilde{x_i}$ is its normalized value, h_i is the j-th HRC axis coordinates HRC_j, and y_i is a binary variable that corresponds to which group the feature belongs.

$$
\begin{aligned}
\max \quad & J_t(X, y) = J_t(h_1(X), h_2(X)) \\
\text{subject to} \quad & \sum_{i=1}^{p} y_i = \lceil p/2 \rceil \\
& y_i \in \{0, 1\}, \text{ for } i = 1, \ldots, p \\
& x \in X, \\
\text{where} \quad & h_1(x) = HRC_1(x) = \sqrt{\frac{\sum_{i=1}^{p} y_i \widetilde{x}_i^2}{\sum_{i=1}^{p} y_i}}, \\
& h_2(x) = HRC_2(x) = \sqrt{\frac{\sum_{i=1}^{p} (1 - y_i) \widetilde{x}_i^2}{\sum_{i=1}^{p} (1 - y_i)}}
\end{aligned}
\tag{5.6}
$$

This formulation is still correct when a dummy variable is used for an odd number of features, as $p = \lceil p/2 \rceil \times 2$. The formulation enables a means to solve the group selection problem while also yielding the visualization. However, with non-linear objectives of $\boldsymbol{h}(X)$, optima are far from trivial. Linear under-estimators [40] or strict psuedo-boolean methods [35] cannot necessarily be used here to simplify or to make the non-linear optimization more efficient. After presentation of various objective functions, this topic is revisited in this research.

In the event of having class labels for each exemplar, *i.e.*, supervised data, a technique related to discriminant analysis can be used. Recall, the Rayleigh quotient in Equation 3.22, where it is maximized in order to best separate projected class means while also minimizing within-class variance. Rather than solving for an optimal projection and projecting the data, instead, the groupings and corresponding visualization coordinates that best separate class means and minimize within-class variance in the hyper-radial space can be solved for directly. Therefore, the intuitive nature of the hyper-radial methodology is maintained without adding the need for the additional interpretation of projections. Let the within-class variance for the visualization be

$$S_W = \sum_{i=1}^{c} \sum_{x \in X_i} (\boldsymbol{h}(x) - \mu_i)^T (\boldsymbol{h}(x) - \mu_i), \tag{5.7}$$

where the subscripts denote the class and μ_i is the mean of the *HRC* coordinates, in row vector form, for exemplars in class i. Let the between-class variance S_B be defined so that the total scatter found in the visualization data is $S_B + S_W$. Therefore,

$$S_B = \sum_{i=1}^{c} n_i (\mu_i - \mu)^T (\mu_i - \mu), \tag{5.8}$$

where μ is the overall mean of the *HRC* coordinates and n_i reflects the number of exemplars in class i [68]. Then the ratio of interest to optimize is

$$J_1 (h_1(X), h_2(X)) = \frac{|S_B|}{|S_W|}, \tag{5.9}$$

where $|A| = det(A) = \prod_l \lambda_l$ and λ_l are the eigenvalues of the matrix A. Thus, this uses the products of the 'variances' in the principal directions, or square of the hyperellipsoidal scattering volume [68]. Again, this is analogous to multiple discriminant analysis (MDA) except that instead of a projection, the solution is for optimality in the visualization coordinates. Unfortunately, unlike MDA where optimality is found via an eigen-problem, optimality for Equation 5.6 using J_1 is not as straightforward. This is discussed shortly.

The solution for J_1 best linearly separates the data. However, this solution is not guaranteed to also spread the data well across the axes. For this purpose, consider the objective,

$$J_2 (h_1(X), h_2(X)) = \frac{tr(S_B)}{tr(S_W)}, \tag{5.10}$$

where $tr(A) = \sum_i A_{ii} = \sum_l \lambda_l$. This objective still tries to separate class means and minimize within-class scatter, but it does so at more of an aggregation across the *HRC* axes by using the sums of the 'variances' in the principal directions.

With p features and assuming two groups, there are $\binom{p}{\lceil p/2 \rceil}$ ways to assign the features into groups. For small p, this makes complete enumeration possible. For example, there

are only 20 total possibilities for the Vertebral Column dataset. The optimal visualization for J_1 on this dataset is depicted in Figure 5.8. There is clear overlap of classes in the visualization, and some level of clustering. This moderately low level of separation may help to explain both the low Fisher ratio for the dataset as well as moderate classification accuracies found in literature [176].

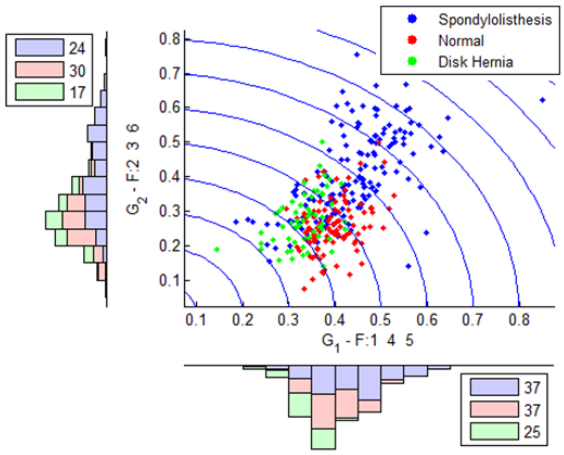
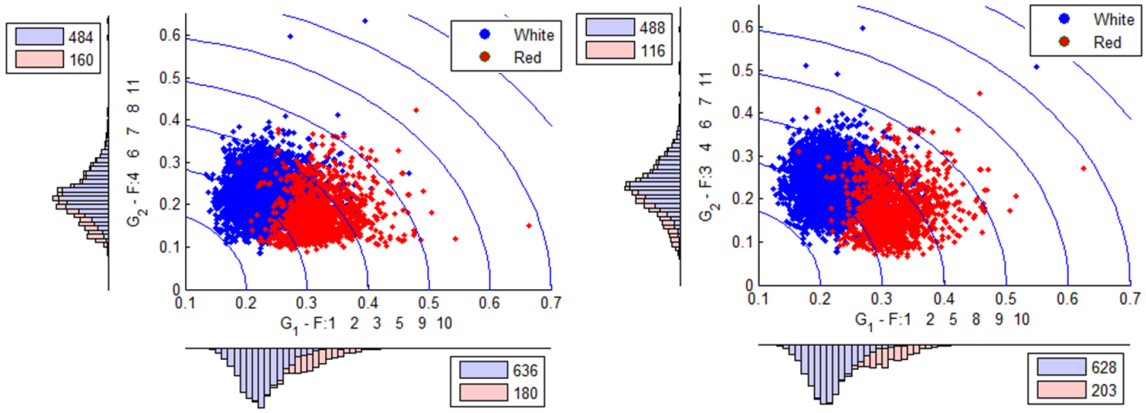


Figure 5.8: Vertebral Column J_1.

The Wine Quality dataset can also be evaluated for true optima as there are only 462 possible groupings. Figure 5.9 depicts the optimal visualizations for J_1 and J_2. Large portions of the White and Red wines separate, with a few obvious outliers. These visualizations also showcase another benefit to the improved HRV visualization, as it is clear that the features in Group 1 form a strong discriminatory group. Here, J_1 and J_2 yield extremely similar group solutions. This is also useful to note as J_1 can take on extremely small function values for certain high-dimensional data, whereas J_2 yields another viable alternative to meet the same intent of class separation, without the possibility of precision issues.



(a) J_1. (b) J_2.

Figure 5.9: Wine Quality.

Class information is not always present for data, and sometimes little is known about the data *a priori*. Therefore, it is also key to investigate unsupervised objectives. In order to help reveal class structure, outliers, and other useful information, three additional objectives are developed with the intent of maximally spreading the data across the visualization. The first technique maximizes entropy over the *HRC* coordinates, where maximal entropy indicates uniformity of the data across the *HRC* axes. First, define a grid of N_g centers over the $[0,1] \times [0,1]$ *HRC* axes. A density d_u is computed using Radial Basis Functions for each grid center u as,

$$d_u = \sum_{x \in X} \frac{1}{\sigma \sqrt{2\pi}} e^{\|h(x)-u\|^2/(2\sigma^2)}, \tag{5.11}$$

In order for these densities to act as probabilities for a larger entropy calculation, a normalization is performed as,

$$\widetilde{d}_u = \frac{d_u}{\sum\limits_{u \in Grid} d_u}. \tag{5.12}$$

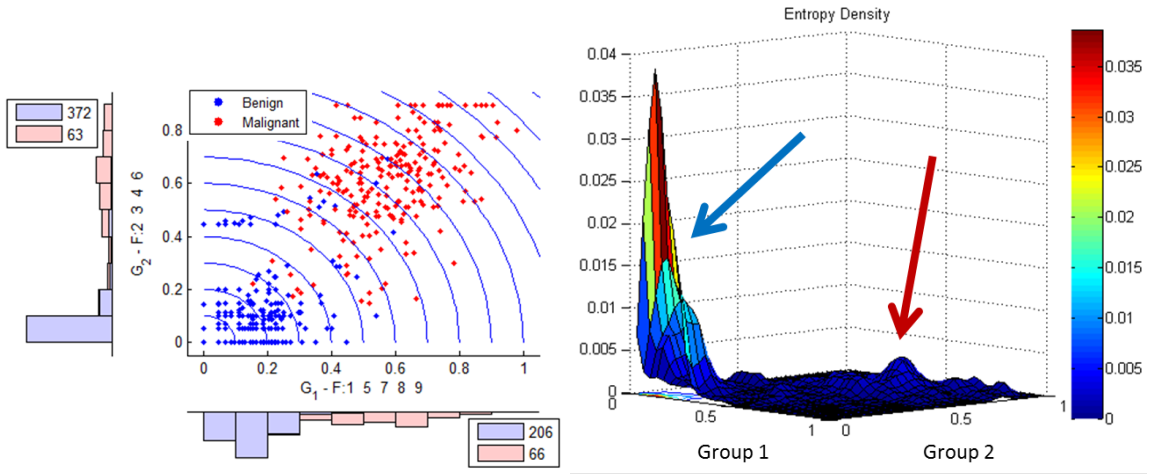
This yields an entropy for the grid as

$$H = -\sum_{u \in Grid} \widetilde{d}_u \ln \widetilde{d}_u, \tag{5.13}$$

where $\ln N_g$ is the maximum possible value of H on the grid. Therefore, H can be scaled by this maximum to form another objective to maximize,

$$J_3\left(h_1(X), h_2(X)\right) = \frac{H}{\ln N_g}. \tag{5.14}$$

Whereas maximizing J_3 seeks the most uniform spread of the data across the groupings, a minimization would seek the most Gaussian representation and could be used to help identify outliers. The apparent limitation of this objective is the choice of σ. Fortunately, it is known that the coordinates are always within the interval $[0, 1]$, no matter the groupings. This makes selection of σ a desired sensitivity that only causes issues if there are many outliers or singleton points in cells. Figure 5.10 shows the optimal J_3 for the Breast Cancer data with $\sigma = 0.025$ and a 33×33 grid. The benign and malignant classes



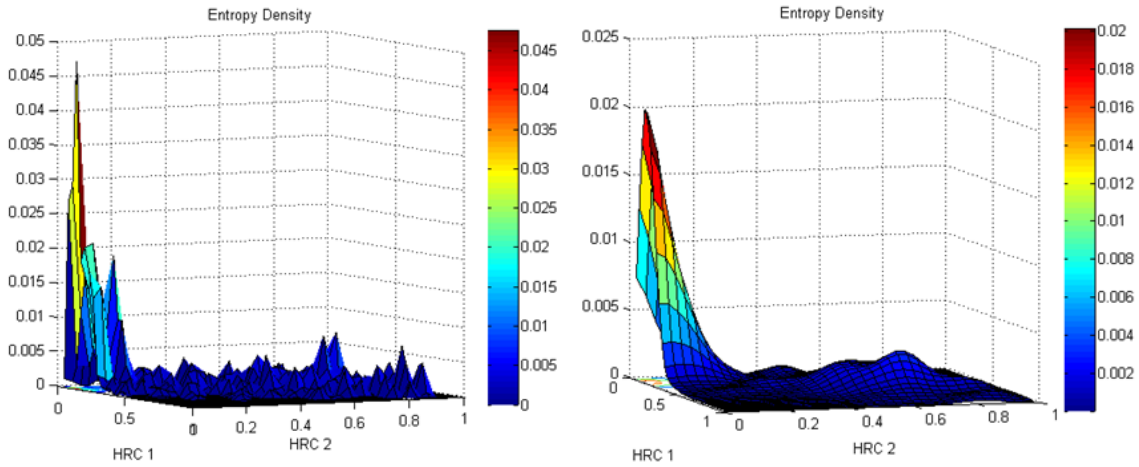
(a) J_3.

(b) Densities.

Figure 5.10: Breast Cancer.

are noticeable by the two separated areas with large density, and in the HRV visualization the presence of two classes is made obvious by the histograms. Figure 5.11 shows the densities with different spread parameters. Although the optimal visualization in this case

would be the same, it can be seen that a $\sigma < 0.025$ identifies too many regions of interest, while a $\sigma > 0.05$ is too smooth. In practice, $\sigma = 0.025$ worked well on data.



(a) $\sigma = 0.01$. (b) $\sigma = 0.05$.

Figure 5.11: Breast Cancer σ Comparison.

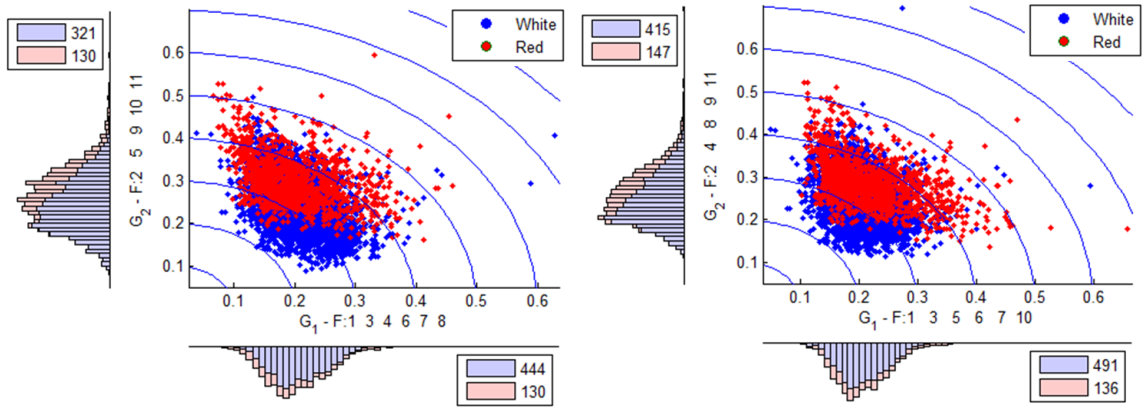
A few other objectives, more simple than J_3, also serve to spread the data as best as possible within the visualization. Such approaches can be particularly useful in trying to identify outliers. Maximizing the absolute value of the correlation between axes would best spread the data along the (h_1, h_2) diagonal, but likely also makes the visualization very linear and makes it harder to see data characteristics. A similar idea is to maximize the combined (h_1, h_2) spread. One way to do this is to multiply the variances found in each direction,

$$J_4\left(h_1(X), h_2(X)\right) = \prod_{i=1}^{2} Var(h_i(X)). \tag{5.15}$$

Another is to force the spread to the extremes in both directions simultaneously while trying to avoid bias in any one direction,

$$J_5\left(h_1(X), h_2(X)\right) = \prod_{i=1}^{2} \left(\max_{x \in X} h_i(x) - \min_{x \in X} h_i(x)\right). \tag{5.16}$$

These objectives are shown for the Wine Quality dataset in Figure 5.12. The groups found are different than with the supervised objectives, with significantly more overlap between classes for this dataset. However, outliers are clear from these visualizations, and the groupings indicate a subset of features that make these outliers so different. With well-separated data, these objectives would also still yield valuable visualizations for class structure exploration.



(a) J_4. (b) J_5.

Figure 5.12: Wine Quality.

All of the objectives presented may yield valuable insight into class structure, outliers, possible clusters, and discriminatory features. Additionally, they can be used in an interative fashion to find small groups of discriminatory features. However, for truly high-dimensional data, techniques need to be developed to solve Equation 5.6 because complete enumeration is likely not possible. To do this, two methodologies are developed here. First, a simple heuristic is proposed to generate an optimal or pseudo-optimal visualization, shown as Algorithm 5.1.

Algorithm 5.1 Local Search with Random Poll

Parameters: m = Max Iterations, q = Mutate Probability

$i \leftarrow 1$, $s \leftarrow \lceil p/2 \rceil$

$y_1, y_2, \ldots, y_s \leftarrow 1$, $y_{s+1}, y_{s+2}, \ldots, y_{s\times 2} \leftarrow 0$

$J \leftarrow J_t(X, y)$

while $i < m$ or until convergence **do**

    $\widetilde{y} \leftarrow y$

    $G_1 \leftarrow \left\{j : y_j = 1\right\}$, $G_2 \leftarrow \left\{j : y_j = 0\right\}$

    $r_1, r_2, r_3 \leftarrow random(0, 1)$

    if $r_3 \geq q$ (Switch features between groups) **then**

        $r_1 \leftarrow \lceil s \times r_1 \rceil$, $r_2 \leftarrow \lceil s \times r_2 \rceil$

        $\widetilde{y}_{G_1(r_1)} \leftarrow 0$, $\widetilde{y}_{G_2(r_2)} \leftarrow 1$

    else (Consider random permutation)

        $R_p \leftarrow$ Random Permutation$(1 : 2s)$

        $\widetilde{y}_{R_p(1:s)} \leftarrow 1$, $\widetilde{y}_{R_p(s+1:2s)} \leftarrow 0$

    end if

    $\widetilde{J} \leftarrow J_t(X, \widetilde{y})$

    if $\widetilde{J} > J$ **then**

        $J \leftarrow \widetilde{J}$, $y \leftarrow \widetilde{y}$

    end if

    $i \leftarrow i + 1$

end while

The heuristic proposed is very similar to a simulated annealing algorithm, with the exception that there is no cooling to allow for movement to a bad solution. Instead, the random poll is used more frequently in lieu of allowing a move to a worse solution. The random poll can also be thought of as a mutation, as an entirely new permutation of features is generated for the two groups. This algorithm is beneficial over many heuristics in that the number of parameters is minimal. Although a genetic algorithm would yield more diversity initially, the random poll arguably provides more diversity over time and this algorithm requires less data storage at any iteration. This could be of particular concern for J_3, and because the objectives are not easily updated when the groups change. Therefore, to maintain efficiency while allowing for some method of escape from local optima, more of a stochastic optimization approach is used. On a non-mutation or non-poll iteration, features are swapped between groups. As with any heuristic, convergence is dependent on the starting iterate and the number of iterations used, perhaps moreso the latter due to the stochastic approach. This is necessary however, as the size of the feasible solution space grows very large as the number of features increases.

As an alternative to Algorithm 5.1, non-linear programming methods can also be used if the binary constraints are relaxed, making Equation 5.6 a highly non-linear objective over continuous variables. In this case, an interior point or active set method, as two examples, can be used on the problem. After solving the relaxed problem, the y_i variables can be set to 0 or 1 based on magnitude such that the largest $\lceil p/2 \rceil$ become 1 and the remaining are set to 0. A comparison of these approaches is discussed in Section 5.3.3

5.3.2 3-Dimensional Hyper-Radial Visualization.

The R from original HRV was not used for the improved two-dimensional HRV visualization. This was for two reasons: 1) it provides redundant information and 2) removal freed a third axis for another *HRC* coordinate. This third coordinate can help to alleviate crowding as the number of classes, features, and exemplars increases.

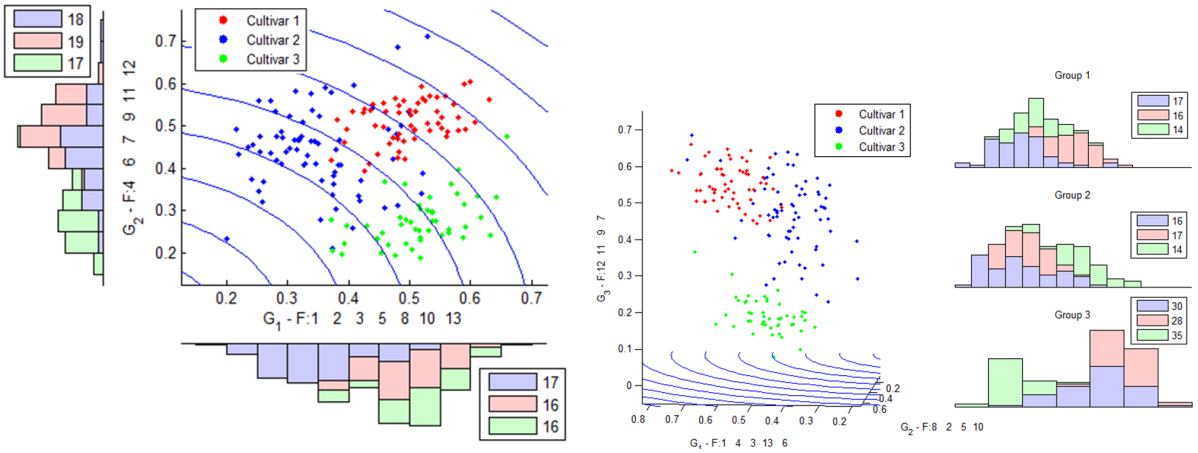
Incorporating a third group is enabled in Equation 5.6 by adding another set of binary variables, and using dummy features as needed to ensure equal group size (and thus, no bias). To expand the formulation to three groups, the binary constraints become,

$$\begin{aligned} \sum_{i=1}^{p} y_i &= \lceil p/3 \rceil \\ \sum_{i=1}^{p} z_i &= \lceil p/3 \rceil \\ y_i + z_i &\leq 1 \quad \text{for } i = 1, \ldots, p \\ y_i, z_i &\in \{0, 1\}, \quad \text{for } i = 1, \ldots, p. \end{aligned} \tag{5.17}$$

The Wine dataset is a good example of the benefit provided by adding the third group. The dataset is generally thought to have well-behaved class structure [19], but with only two groups there is still class overlap in the visualization. However, using a third group distinguishes the class boundaries better, as shown in Figure 5.13. In both cases, these visualization were found using Algorithm 5.1 with $m = 8500$ and $q = 0.4$. The three-dimensional visualization suggests that the third cultivar can be largely distinguished by the seventh, ninth, eleventh, and twelfth features. Figure 5.14 provides a comparison to the t-SNE visualization and the first three major components from PCA. As can be seen, the improved HRV provided a clearer distinction between classes.

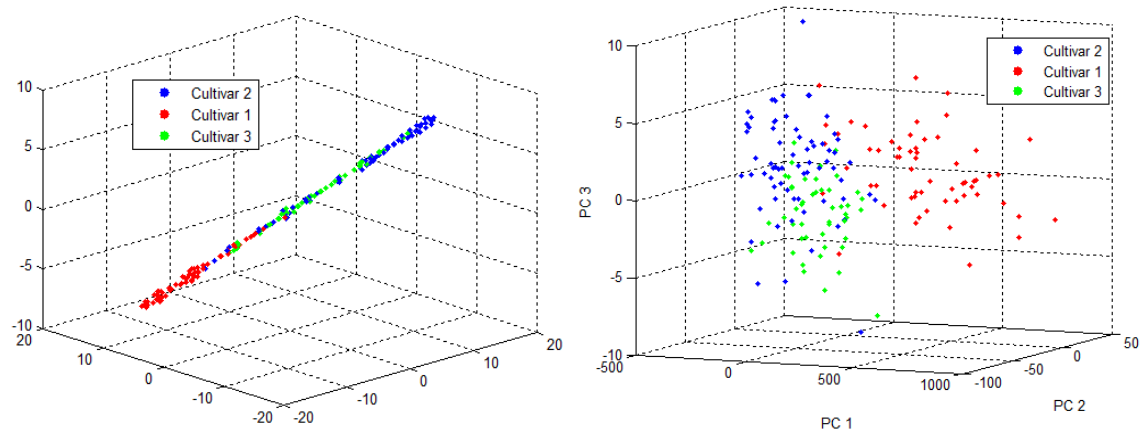
The MNIST dataset has a much larger number of features. With all ten classes included, the visualization is still too crowded, but consider the full training 0 and 1 classes. Removing pixels that are zero for all exemplars from both of these classes, the number of possible groupings is still $\binom{617}{206\ 206\ 205} = 617!/\left((206!)^2\, 205!\right)$. Figure 5.15 depicts two solutions solving the relaxed problem with an interior point method.

Using the J_1 objective, the 1-digits present distinctly lower across Groups 1 and 3, and it is obvious that there are two classes present even without class information in the visualization. Using J_2, Group 1 is highly discriminatory and the 1-digits present distinctly



(a) Two Groups. (b) Three Groups.

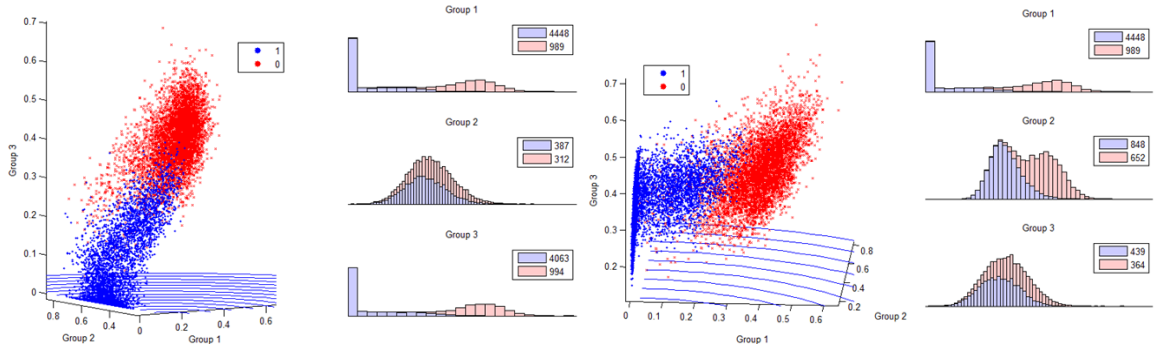
Figure 5.13: Wine J_1.



(a) t-SNE. (b) PCA.

Figure 5.14: Wine Visualizations.

lower in Group 2. The heuristic with a similar number of iterations to that used by the interior point method, $2000 \leq m \leq 4000$, typically provided a J_1 visualization that did not discriminate as well in the pure visual sense, and a J_2 visualization that was very similar to



(a) J_1. (b) J_2.

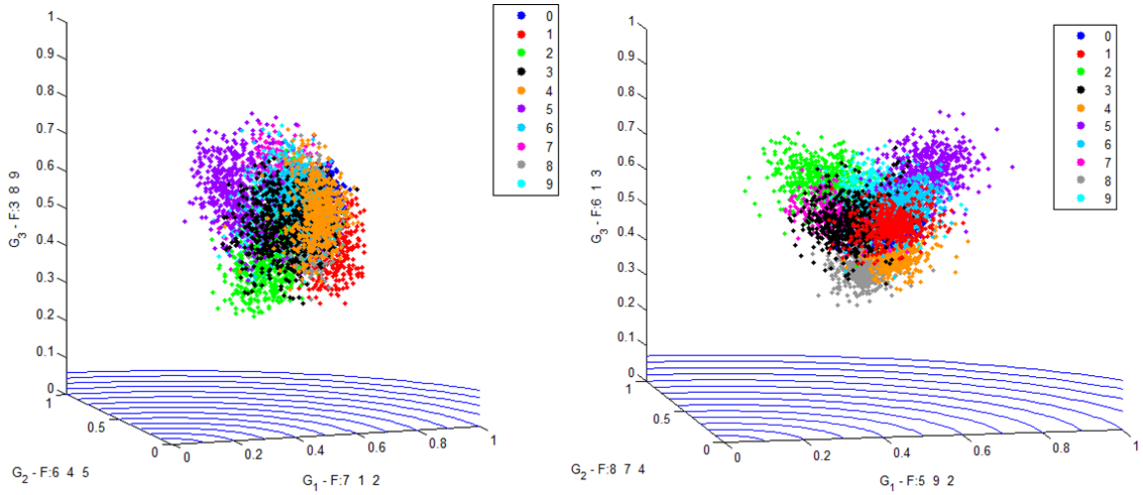
Figure 5.15: MNIST.

Figure 5.15(a). In the unsupervised case using the heuristic, J_5 typically yielded the same visualization as Figure 5.15(a).

Previously, data projections were discussed both as visualizations themselves and inputs to visualizations. As MDA provides the optimal linear linear projections for class separation, the MDA component scores should visualize better in HRV than PCs scores if the improved HRV is a 'good' visualization. Using a random sample of 600 exemplars from each class in MNIST for sparsity purposes, Figure 5.16 shows the J_1 optima on the PC and MDA scores for the nine major components in each case, where here the histograms are not included so as to provide more space. The PC scores are more compact and have significant overlap in any direction, while the MDA scores break out nicely by class. Further, the MDA scores provide a better geometry that in the unsupervised setting might suggest the presence of multiple classes. This is a validation of the improved HRV visualization.

5.3.3 Further Visualization Analysis.

The improved HRV visualization can also be applied to HSI data after it has been reshaped to pixel-by-band. First, consider Figure 5.17. Here, all target pixels and an additional random sampling of background pixels from ARES1D were chosen, for a total

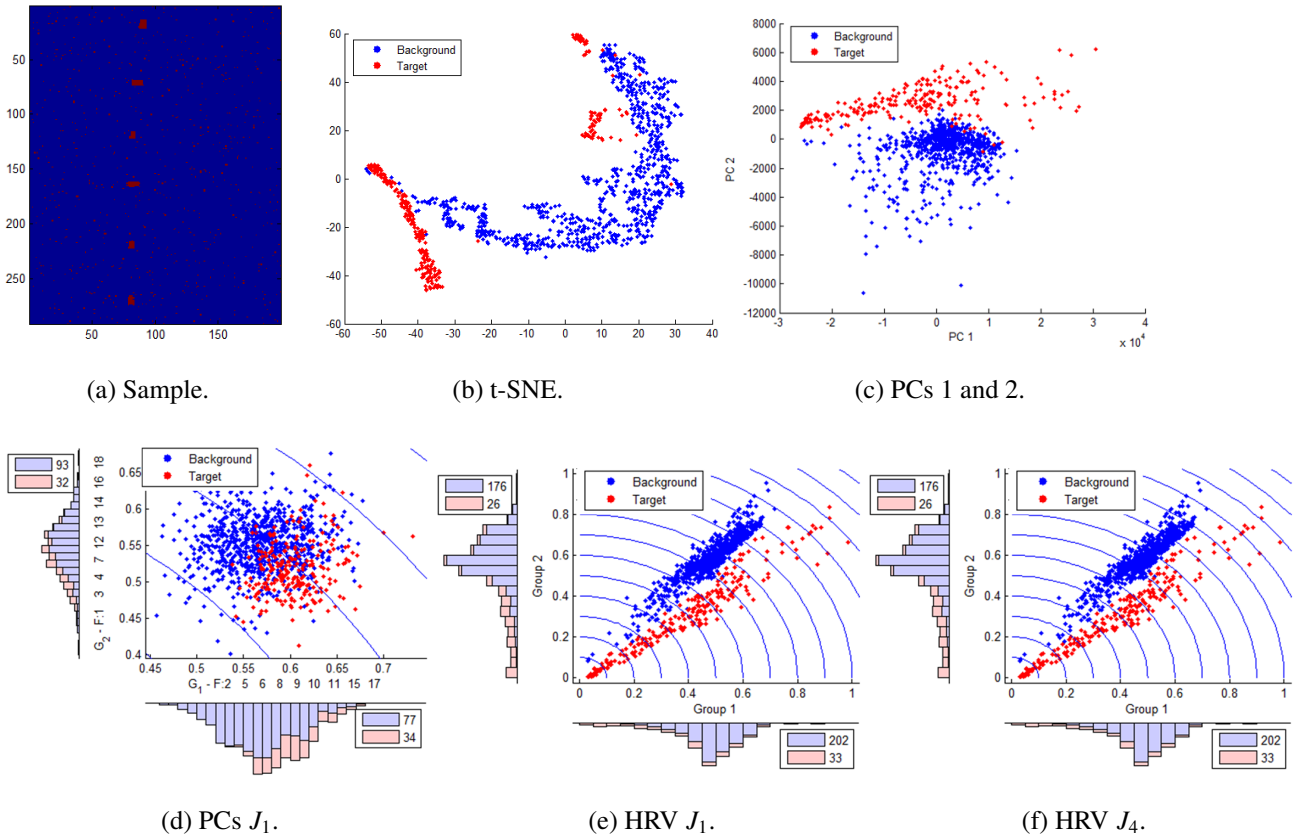


(a) PCA. (b) MDA.

Figure 5.16: MNIST J_1.

of 1,000 pixels. This sample is shown in Figure 5.17(a). t-SNE on this sample separate the targets and background fairly well, but not as well as HRV using J_1 or J_4 where there is clear linear separation between the two classes. Figure 5.17(c) shows that the first two PCs also nearly separate the data, but not as well. Using the first 18 PCs from the MDSL cut-off and their scores, the separation becomes far less obvious within the HRV visualization. This showcases the strength of the improved HRV method and its ability to maintain desirable properties of the data.

Now, recall from Section 4.3.1 the various complexities of the HSI images. From the metrics that were shown, it was implied that ARES2D has more class overlap and is more complex than ARES1D. The J_1 visualizations for these images are shown in Figure 5.18. It is clear from these visualizations that ARES1D is more linearly separable than ARES2D, as suspected, although a subset of the targets in ARES2D are significantly different than background.

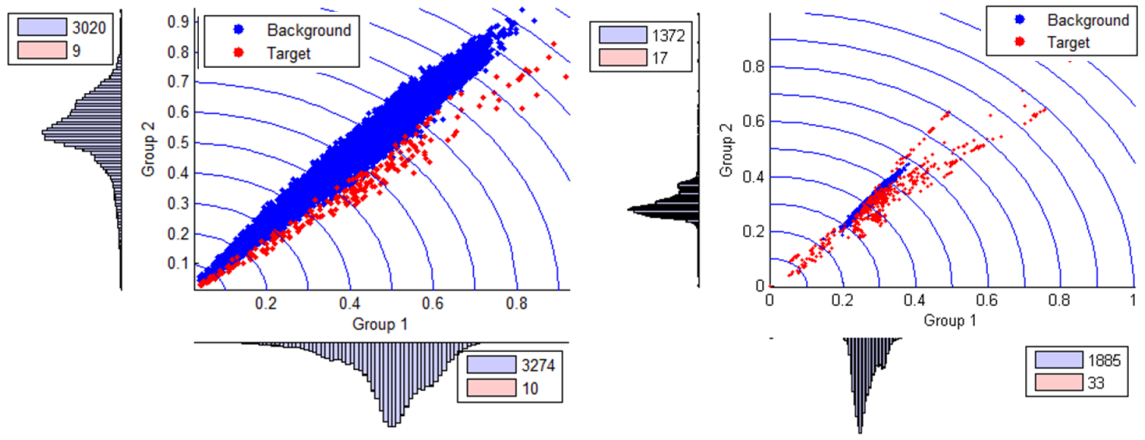


(a) Sample. (b) t-SNE. (c) PCs 1 and 2.

(d) PCs J_1. (e) HRV J_1. (f) HRV J_4.

Figure 5.17: ARES1D Visualization.

At this juncture, further investigation of Algorithm 5.1 and the relaxation seems warranted. For the heuristic, an experiment with 30 replications of each setting was conducted, where m, q, the number of groups (2 or 3), the objective, the dataset, and the starting iterate were varied. The starting iterate was selected based on the original order of features or by calculating the extended Fisher ratios for each feature, choosing that with the highest ratio remaining, and grouping it with all other features that were absolutely correlated 0.8 or higher. The intent for this alternative starting iterate was to guess at which features would provide good discriminatory groups. The Wine Quality, Wine, and MNIST data sets were used for the experiment, where only the 0 and 1 classes were used



(a) ARES1D. (b) ARES2D.

Figure 5.18: ARES1D and ARES2D Comparison.

for MNIST. m was varied within $[500, 10,000]$ and q was varied within $[0, 1]$. In general, Algorithm 5.1 showed better objective values when q was non-zero and as m increased, and was fairly efficient in converging to local maxima. The alternative starting iterate did not conclusively provide better solutions.

For the relaxed version of the formulation, the objective, starting iterate, number of groups, optimization algorithm, and method of setting the y_i after solution were varied across the three data sets. Both an interior point method and active set method were used to solve the non-linear program, so as to achieve at least local maxima. The y_i variables were set to 1 and 0 by sorting the final relaxed solution by magnitude and splitting them into appropriate groups, or by applying the same methodology at the end of each iteration of the optimization process. As one might hope, this methodology had little impact on either the solution or the efficiency. Further, the y_i only sometimes broke into distinct groups once sorted by magnitude, where other times the distribution was very linear in nature. These solutions may have suggested local maxima that could be far from the global maximum, or

at least that the technique of taking the largest variables and putting them into one group may not always be the best solution. However, solving the relaxed problem proved to be highly efficient as well despite the presence of so many variables, is less random, and found better solutions than the heuristic in some cases. The active set method provided better solutions for the three-group case with J_1, but otherwise the interior point method provided slightly better solutions.

Unfortunately, deeming one method or setting entirely better than another proved extremely difficult during analysis due to the variability. The heuristic often found better values for the J_1 objective on the data sets, but the interior point method found a few better solutions, and more consistently, for the J_2 objective. A table with best objective function values found for J_1 and J_2 specifically is shown as Table 5.2 for lack of a better analytic synopsis. The best values found for the Wine Quality and Wine data sets were, in fact,

Table 5.2: Algorithm Comparison.

Dataset (Number of Groups)	Heuristic - J_1	Relaxed - J_1	Heuristic - J_2	Relaxed - J_2
Wine Quality (2)	1.19×10^{-9}	1.37×10^{-10}	1824.72	1824.72
Wine (2)	6827.86	6100.36	119.75	119.75
MNIST (2)	4.36×10^{-8}	1.6×10^{-9}	4654.09	9765.89
Wine Quality (3)	3.89×10^{-22}	8.7×10^{-27}	1788.25	1290.06
Wine (3)	4.85×10^{-10}	1.73×10^{-11}	120.81	116.53
MNIST (3)	9.78×10^{-20}	6.2×10^{-21}	6187.81	11918.34

optimal. Again, no optimal settings or trends beyond those already discussed were evident from the experiment. Improvement in the objective from the starting iterate was always observed, however.

In general, the visualization methodologies proposed work best with a moderate number of features and a few classes. As with any visualization, its capability is constrained by the amount of feature information that can be maintained in a low dimensionality. However, improved HRV seems very useful and intuitive in identifying data outliers, structure, discriminatory feature groupings (even dynamically), comparing transformations, and comparing potential classification complexity. The technique is very simple and does not change the inherent properties of the data, making it easy to interpret. Additionally, the visualization is computationally efficient given the formulation and solution methodologies presented.

With band removal, truth mask analysis, and now visual interpretation of the HSI data achieved, focus is put primarily on developing anomaly detection algorithms. The improved HRV is used again in Chapter 7 to contrast data skeletons, but first, a global anomaly detection framework is developed.

VI. Factor-Based Global Anomaly Detection

Johnson's AutoGAD [111] and Jablonski's MPCA [107] algorithms have been shown to perform extremely well on certain HYDICE imagery. In fact, Jablonski [107] also showed that re-defining one of Johnson's parameters could greatly increase AutoGAD's TPF, although it also increased the FPF. Both of these methods rely heavily on measures generated from principal components, where the resulting mappings are not easily interpreted. Due to their high performance, these techniques are used as the primary comparisons for methods developed here.

Given the strong performance of the factor model in Chapter 4 for purposes of band selection, investigation into using factors within related frameworks is warranted. Whereas PCA yields the direction of principal variance in the image, factor analysis with a rotation groups the variables according to a predetermined notion of structure. This enables more direct interpretation of the components, while not affecting reconstruction error. Ideally, these components reveal better or more efficient mappings with which to identify anomalies, as certain bands and areas of the spectrum are grouped together.

Before proceeding, recall that in this research a slightly different set of bands are being used for the HSI analysis than in the originating AutoGAD and MPCA research. Experimentation showed that this had little impact on the TPF and FPF rates for the AutoGAD and MPCA algorithms at their published optimal settings. Also, recall that border pixels in ARES images that have such pixels in their truth masks, are treated entirely as background due to the insights from Chapter 4. As discussed previously, in general, this reduces TPF rates and increases FPF rates in comparison to the originating AutoGAD and MPCA research. In those cases, border pixels were used in the numerator for the TPF and denominator for the FPF. Thus, such rates in this research are more conservative.

This Chapter begins by discussing the AutoGAD and MPCA algorithms and investigating direct application of Factor Analysis (FA) into those methodologies. A technique is also established for experiments on the respective algorithms. Next, specific areas of those algorithms are discussed and experimented with in more detail, in order to help shape a refined framework in which to use FA. Finally, the resulting framework is analyzed and optimized to provide a new global anomaly detection algorithm for HSI.

6.1 Existing Component-Based Global Anomaly Detection

The AutoGAD and MPCA algorithms, previously presented in Sections 3.11.3 and 3.11.4, are used as primary points of comparison and motivation for the algorithms developed here. The AutoGAD and MPCA have several concepts in common, and neither provides mappings that are directly interpretable. The AutoGAD algorithm begins by converting the major principal components (PCs) into independent components. This is done via Fast ICA, and is not deterministic. From the set of resulting components, those with a maximum score above a threshold and potential anomaly (PA) SNR above another threshold are selected for IAN filtering. The PA SNR is calculated by finding the first zero count bin from the center in a histogram of the component scores, and splitting the pixels into background and potential anomaly based on this location. The PA SNR is defined as,

$$PA\ SNR = 10\log_{10}\frac{\mathrm{var}\left(potential\ anomalies\right)}{\mathrm{var}\left(background\right)}. \tag{6.1}$$

Nominated components are filtered using a moving smoothing window, or neighborhood, of size $w \times w$ pixels. For each pixel's neighborhood, a mean and variance estimate are taken and the current pixel score is replaced with a filtered score,

$$score_{new} = \mu + \frac{\sigma^2 - v^2}{\sigma^2}\left(score_{orig} - \mu\right), \tag{6.2}$$

where μ is the neighborhood mean, σ^2 is the neighborhood variance estimate, and v^2 is the average of the neighborhood variances, treated as an estimate for the system noise variance [110]. Depending on how many times this is done, it can significantly affect the mapping,

and is used to better separate anomalies from background. It is advantageous in that it provides a spatial element the algorithm. After filtering, each component is thresholded again using the first zero-bin histogram in order to determine anomalous pixels. Both negative and positive scores can be thresholded and evaluated by performing the zero-bin detection on each side of the center of the scores. After experiments on both the HYDICE and AVIRIS imagery, the author can confirm that the ICA step is important to success of this algorithm, in comparison to only using the PCs.

Alternatively, MPCA builds four new, aggregate components from the IAN filtered PCs, using reconstruction error, sums of the major components and minor components, and medians. These new four components are then filtered themselves, and thresholded based on the first zero-bin histogram method. Potential anomalies are removed from the covariance estimate that was used to build the original PCs, and the process occurs again, where on the second iteration, PA SNR is used to remove components that may yield high false positives. Upon completion, a vote is taken of the remaining components in order to derive anomalies.

Both AutoGAD and MPCA use a large number of parameters. Optimal settings for AutoGAD, based on a RPD for several HYDICE images, are shown in Table 6.1. Jablonski replaced the bin width parameters b_{SNR} and b_i for the zero-bin histogram steps with a re-defined single bin width size parameter Y that adjusted the width to the range of scores for the component and number of pixels in the image N, such that this bin width ω was found for any mapping by,

$$\omega = \frac{Y}{N} \left(\max(scores) - \min(scores) \right). \tag{6.3}$$

Using $Y = 300$, he found higher TPF rates than the RPD optimal settings on the same imagery, but also increased FPF rates [107]. This was due to zero-bin detection becoming more sensitive on certain components and less sensitive on others. Larger values of Y

Table 6.1: Base AutoGAD Parameter Settings [111].

Parameter	Name	Setting
c_k	MDSL Dimension Adjust	-1
b_{SNR}	Bin Width SNR	0.03
t_{SNR}	PA SNR Threshold	3
t_{MS}	Max Score Threshold	9
t_l	Low PA SNR	10
I_h	IAN Filtering Iterations (High SNR)	100
I_l	IAN Filtering Iterations (Low SNR)	20
w	Window Size for IAN Filter	3
b_i	Bin Width Identify	0.06

yielded even higher TPFs, but began to greatly inflate the FPFs. Due to the high margin of improvement on the TPF rates for some images such as ARES4F and ARES2F (> 0.2 increase), this version of AutoGAD with $Y = 300$ and the Table 6.1 settings otherwise, is used as a base comparison in this research.

In order to optimize the MPCA algorithm, Jablonski fit a second order model without interaction terms to a three-level full-factorial design using the response,

$$(\mu_{TPF} - 1)^2 + 3\mu_{FPF}^2 + 3\sigma_{FPF}^2, \tag{6.4}$$

where this was a function of mean TPF rate, mean FPF rate, and sample variance for the FPF rates across a training set of images [107]. This was designed to generate low false alarm rates. His resulting optimal settings for MPCA are shown in Table 6.2.

When validating MPCA on images with no targets, originally he saw a higher FPF. For example, ARES1C had a 0.106 FPF and ARES2C had a 0.073 FPF. To negate this, he thresholded Johnson's Potential Anomaly SNR measure from AutoGAD [110], Equation

Table 6.2: Base MPCA Parameter Settings [107].

Parameter	Name	Setting
c_k	MDSL Dimension Adjust	-4
$Y_{initial}$	Initial Detection Sensitivity	0.249
Y_{final,D_1}	D_1 Final Detection Sensitivity	3.5
Y_{final,D_2}	D_2 Final Detection Sensitivity	2.774
Y_{final,D_3}	D_3 Final Detection Sensitivity	2.856
Y_{final,D_4}	D_4 Final Detection Sensitivity	2.295
$I_{D,23}$	$D_{2,3}$ IAN Filtering Iterations	8
$I_{D,4}$	D_4 IAN Filtering Iterations	2
I_{pc}	PC IAN Filtering Iterations	2
w	Window Size for IAN Filter	3

6.1, on the components D_i, where the background and potential anomalies were again chosen by the first-zero-bin histogram method. This threshold improved MPCA's FPF on test and validation images, and included reducing the FPF for ARES1C and ARES2C to 0. After experiments in this research, it was confirmed that it also affected his training images' rates only slightly, with the exception of the ARES1D image. Using the optimistic truth mask as he did in his research (*i.e.*, treat border pixels as targets if they are declared targets), the ARES1D TPF reduced from 0.988 to 0.926 while the FPF reduced from 0.043 to 0.029. As it turns out, this is partly due to the homogeneity of the target pixels in some of the primary components for the ARES1D image, thus yielding a low PA SNR. Baseline MPCA results shown in this chapter include the SNR threshold.

AutoGAD and MPCA are of such high interest, because factor analysis may provide more interpretable components than these existing methods and ideally better separates

background and anomaly as a result. The groupings of bands found after rotation, assuming that the fundamental assumption of anomalies being significantly different than background holds, should yield mappings that reveal where in the spectrum anomalies are different than background. PCA does this in part, but the components are entirely based on variance distribution. Varimax rotated factors take these components and highly load the bands onto respective factors. Specifically, the Varimax rotation maximizes the variance of the loadings, while constraining the factors to be uncorrelated [115]. Thus, where anomalies are significantly different in radiance on the spectrum, because materials reflect differently in different areas of the spectrum, should be revealed by respective factors. As a result, these factor mappings may provide better distinction between background and anomaly, and representations of the materials found in the image. In the case of AutoGAD, an additional advantage is that if factors could be used instead of independent components, this increases the efficiency of the algorithm and removes the random component of its performance.

6.2 Component Generation and Selection

Investigation can begin by evaluating components at different stages of the existing algorithms, and by evaluating versions of AutoGAD where ICA is not performed, or only PCA or Factor Analysis (FA) are performed within a similar framework. Table 6.3 shows the maximum two-class Fisher ratio (Equation 3.81) of components for different stages and versions of each algorithm, comparing full pixel targets to background, for many of the HYDICE and AVIRIS images. Here, the PCA, ICA, and Factor Analysis (FA) rows depict those variations of AutoGAD, where the respective components are used. For factor analysis, the Varimax rotation is applied. The ICA case is a mean of twenty repetitions of standard AutoGAD, where there was negligible variability. The *Thres* rows of the table list the maximum ratio among only those components meeting the maximum score and PA SNR thresholds from Table 6.1, *IAN* denotes after the IAN filtering iterations, and MPCA

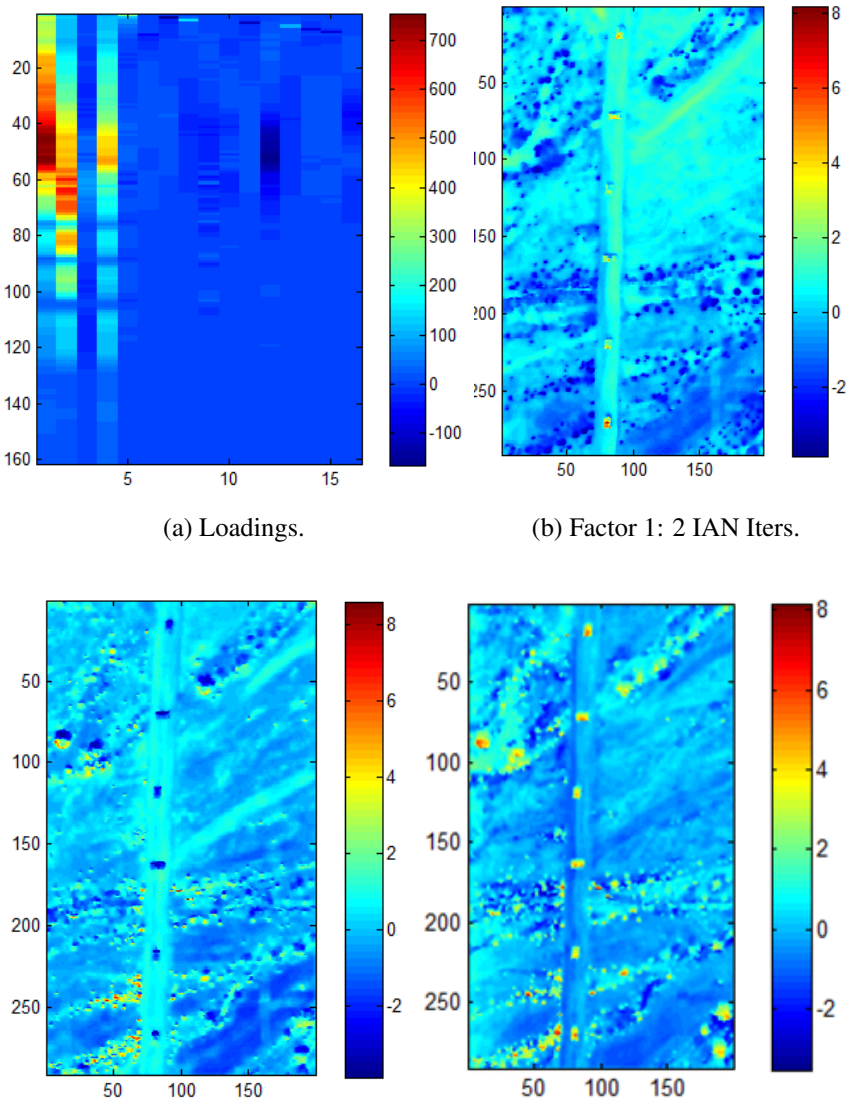
is split into before and after removal of potential anomalies towards constructing the D_i components. The mean and standard deviation of all components under consideration were also computed for each of these variants, but did not provide a great deal more information.

	ARES1F	ARES3F	ARES4F	ARES1D	ARES2D	4Ships2	Scene1	Ship1	Virgin1
PCA	6.9183	1.2811	0.5170	3.5523	0.8532	1.9529	0.3867	2.1709	2.5081
ICA	0.7148	1.4143	0.3907	3.2807	0.2677	0.6939	0.6139	1.3335	0.8389
FA	1.9497	0.9104	0.8194	5.8328	0.6819	1.0628	1.7482	0.6731	2.3863
PCA (Thres)	6.9183	1.2811	0.1778	3.2799	0.2382	1.9529	0.3867	2.1709	2.5081
ICA (Thres)	0.7148	1.4143	0.3907	3.2807	0.2677	0.6939	0.6139	0.8299	0.8389
FA (Thres)	1.9497	0.1280	0.8194	--	0.3027	1.0628	1.7482	0.6731	2.3863
PCA (IAN)	3.7257	0.4826	0.1257	1.6376	0.1421	2.3189	0.3688	2.0242	1.7315
ICA (IAN)	0.8844	1.2131	0.2990	5.6860	0.2657	0.7152	0.6121	0.8666	0.8616
FA (IAN)	2.4513	0.2147	0.4827	--	0.3030	1.1358	1.8031	0.5706	0.8529
MPCA (1st Iter)	3.5050	1.3569	0.9723	2.8872	2.3287	2.4130	1.9107	1.6300	1.9362
MPCA (Final)	2.7890	1.2764	0.9344	2.7744	1.8027	1.5127	0.0193	0.6512	1.5304

Table 6.3: Max Component 2-Class Fisher Ratio.

Table 6.3 provides useful insight. First, the separation of target and background on any single component is not critical to the success of AutoGAD or MPCA. Rather, the fusion of information across several good components is more important. For instance, based on experimentation it was found that MPCA truly has better results after the second iteration, yet the Fisher ratios are all lower after the iteration. This suggests components that represent different characteristics of the image would be most useful, perhaps such as those found by FA. The fact that these ratios were not always higher after IAN filtering for AutoGAD with ICA also supports this. The benefit of filtering was shown when AutoGAD's parameters were optimized by Johnson, yet lower Fisher ratios suggest that only certain target pixels are being accentuated for different components.

Before investigating these algorithms further, consider the ARES1D, ARES1F, and ARES4F images specifically. Figure 6.1 depicts IAN filtered factor scores for the three factors with the highest loadings for ARES1D. The first factor seems to represent the



(a) Loadings.

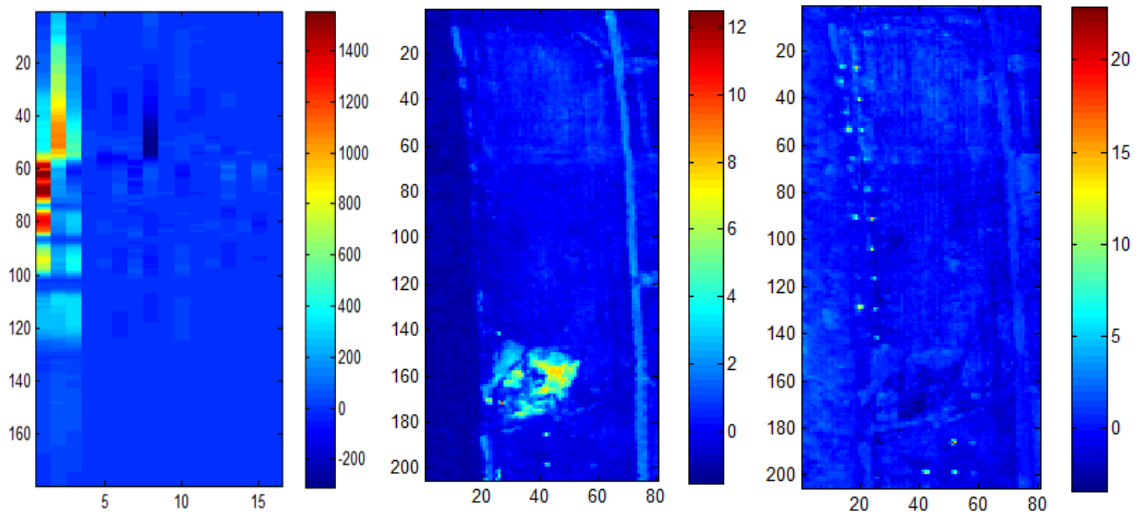
(b) Factor 1: 2 IAN Iters.

(c) Factor 2: 10 IAN Iters.

(d) Factor 4: 2 IAN Iters.

Figure 6.1: ARES1D Factors.

background in total, with full pixel targets revealed on the mapping. The second factor reveals brush, and the third reveals vehicles and brush. Thus, these each represent different materials in the image. The IAN filtered versions are shown here so as to simultaneously reflect how increasing smoothing affects mappings. At first, it is tempting to think that factors with the very highest loadings are those desirable to use in order to find anomalies. However, in practice this seems to be very much a function of the image. For example, with ARES4F and ARES1F, the best discriminating factors for anomalies do not have the very highest loadings. Figure 6.2 shows the loadings matrix (band-by-factor), and scores for two of the ARES4F factors before any filtering is applied. The second factor

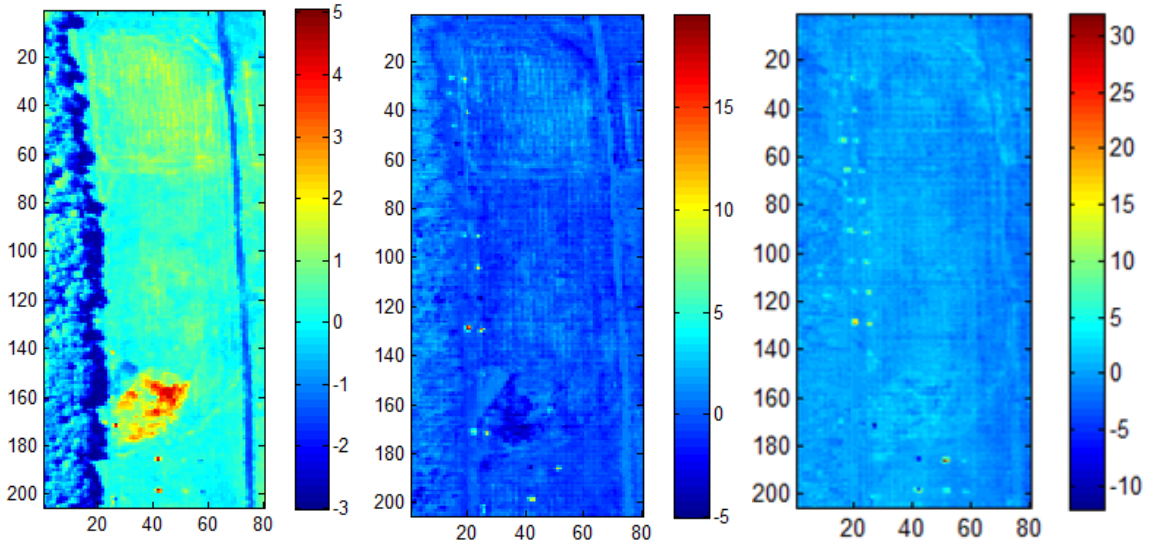


(a) Loadings. (b) Factor 2. (c) Factor 6.

Figure 6.2: ARES4F Factors.

reveals a material in the image and a few targets, while the sixth represents all background materials, thus yielding many more of the targets. When compared with three of the best discriminating PCs from that image, shown in Figure 6.3, it is clear that the factor scores

are more meaningful. The PCs represent a mix of materials in the image, and have more pixels at both ends of the score range or near many target pixels within a small interval of the score range.

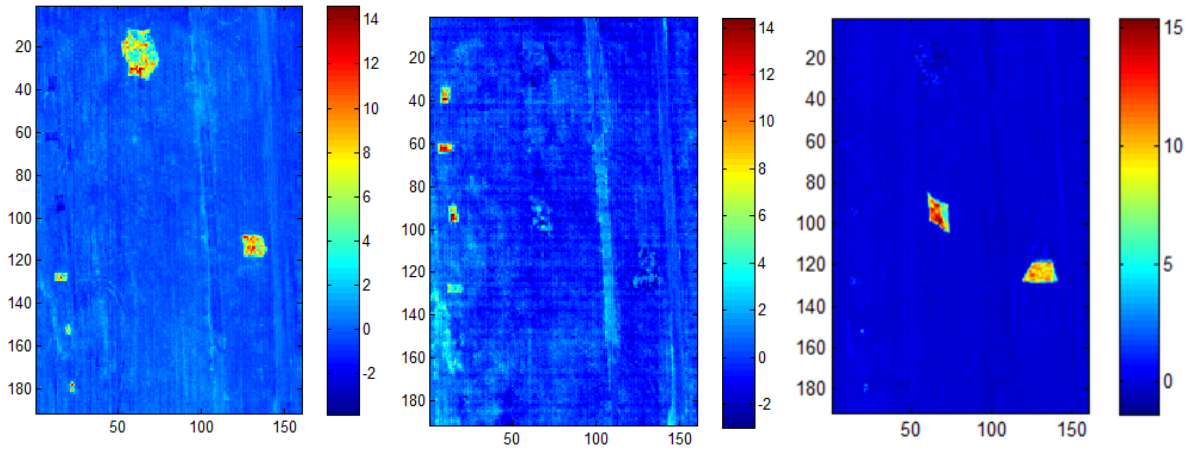


(a) (b) (c)

Figure 6.3: ARES4F PCs.

This does not imply that all factors provide great discrimination of materials. Some are still noisy and yield little discrimination, thus describing regions of the spectrum where many of the materials in the image are similar. However, in practice, various materials of each image were consistently represented by factor mappings within a MDSL cut-off set of factors. This also suggests that the factors could be used to estimate the number of materials, or endmembers, in an image. Figure 6.4 depicts three of the better discriminating ICs for ARES1F (for a given ICA run), excluding two where the road and only some of the targets were revealed. Provided for contrast is one of the factors before any smoothing is applied, shown in Figure 6.5. The factor represents the entire background and reveals all

targets, and admittedly a slight amount of vegetation on the edge, while a combination of the ICs must be used to find all targets.



(a) (b) (c)

Figure 6.4: ARES1F ICs.

Factor analysis is promising, as even the Varimax rotation is not that computationally expensive and the mappings shown discriminate well. However, the same problem as other methods exists. A methodology needs to be used in order to find the specific factors that can discriminate target from background, *i.e.*, the background factors. Further, these factors may need to be smoothed or adjusted so as to make it easier to autonomously identify the potential anomalies. In this research, it is desirable for such a process to be entirely unsupervised. Aside from those methods used by Johnson and Jablonski, a few others exist in the literature for an unsupervised image. Gu, Liu, and Zhang [84, 145] used the Local Singularity rule, a threshold on kurtosis and skewness within local windows, to pick best PCs for RX. This made sense in such a context, as anomalies would break the RX Gaussian assumption for a window. Unfortunately, this does not work if applied globally outside of

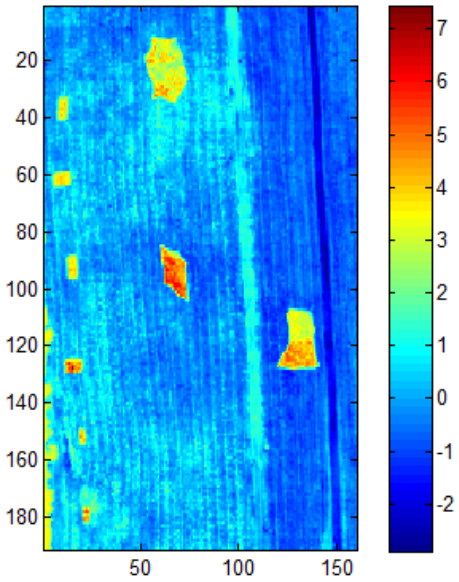


Figure 6.5: ARES1F Factor.

RX, as it is already known that HSI images are typically not Gaussian. They also cited the Minimum Noise Fraction rule, but such a rule requires an estimate on the noise. Given the strength of the maximum score and PA SNR rules utilized within AutoGAD and MPCA, it makes sense to also try and apply them to the factors.

Relatedly, brief investigation into generating new features from the factors is warranted. Kotwal and Chaudhuri [127] developed an optimization process to find the best linear combination of spectral bands and pixels in order to visualize an HSI image. They optimized a function of entropy and variance, where each band had a set of coefficients to optimize for each pixel. At first glance, this technique may seem useful to also generate a new fusion feature for anomaly detection, somewhat akin to MPCA. However, in modifying this technique to optimize over the factors instead of the bands, and using only one coefficient per band, this optimization becomes nearly equivalent to finding a PC or IC.

The component ideas from MPCA can also be applied directly to the factors from FA, yielding a multiple factor analysis algorithm. The reconstruction component D_3 would be the same, as the factors are just rotations of the PCs. The D_1 component can be applied directly to the factor scores. The D_2 and D_4 components can either remain as they were as a function of the minor PCs, or a larger factor model can be assessed and these components can be estimated based on the minor factors of the factor model. This idea, smarter use of maximum score and PA SNR, and factor analysis within AutoGAD are all explored next.

6.3 Direct Application of Factor Analysis

Algorithm 6.1 shows the AutoGAD-like factor analysis framework used in these initial experiments. Note, this is very similar to AutoGAD with FA replacing ICA. FA is performed using the covariance matrix and the unweighted least squares solution for the scores, as presented in Section 3.4. An initial smoothing was added, and maximum scores were taken before any smoothing. The bin width parameters were set without using the range of scores, setting the bin width to a pixels per bin Y divided by the number of pixels in the image. Both negative and positive scores were thresholded when assessing for potential anomalies.

Table 6.4 shows TPF and FPF results for several variations of the AutoGAD and MPCA algorithms on a set of ten images. Again, these TPFs and FPFs treat any border pixels as background. Best TPF and FPF values are highlighted by image. The AutoGAD (ICA) column is a mean of 20 runs, and using $Y = 300/N = b_{SNR} = b_i$ for a common bin size parameter within the first zero-bin detection algorithm. The first two MPCA columns are Jablonski's full MPCA algorithm with the value used for $Y_{Initial}$ in parentheses. The MPCA (Simple) column uses the simplified form without the minor nuances he added to greater avoid false positives, *i.e.*, Algorithm 3.3. All parameters were set to those from Table 6.1 and Table 6.2 unless otherwise denoted. Of note is that focusing on full-pixel targets for the TPF greatly reduces the MPCA TPF on certain problems. For example, the

Algorithm 6.1 Test FA AutoGAD

Remove absorption/noisy bands and reshape the data cube to $N \times p$.

$X^c_{N\times p} \leftarrow \left(X_{N\times p} - \mathbf{1}_{N\times 1}\boldsymbol{\mu}^T\right)$: data is centered.

$b_i = b_{SNR} \leftarrow Y_{ID}/N = Y_{SNR}/N$.

Find eigenvectors V and eigenvalues Λ from $cov\left(X^c_{N\times p}\right)$: do PCA.

Use MDSL to determine the dimensionality k. $L_{p\times k} \leftarrow V_{p\times k}\Lambda^{1/2}$ denotes the factor loadings.

Varimax rotate the loadings to yield $\hat{L}_{p\times k}$.

Compute the factor scores: $F_{N\times k} \leftarrow X^c_{N\times p}\hat{L}_{p\times k}\left(\hat{L}^T_{p\times k}\hat{L}_{p\times k}\right)^{-1}$.

For any $1 \leq i \leq k$:

if $|\min(F^i)| > \max(F^i)$ **then**

    $F^i \leftarrow -F^i$.: Negate the i-th set of factor scores.

end if

For each factor mapping F^i, $m_i \leftarrow \max(F^i)$.

$snr_i \leftarrow PA\ SNR(F^i)$.

Retain F^i if $m_i \geq t_{MS}$ and $snr_i \geq t_{SNR}$, otherwise discard this factor.

$F^i \leftarrow IAN(F^i)$, with I_l iterations if $snr_i < t_l$. Otherwise use I_h.

for $r \in \{-1, 1\}$ (Check both positive and negative scores) **do**

    Define $\eta_i \leftarrow PA\ SNR(r \times F^i)$ as the threshold from the first-zero bin histogram, using scores separated into background and potential anomalies.

    if $r \times F^i_j > \eta_i$ **then**

        Declare pixel j anomalous.

    end if

end for

TPF for ARES4F reduced from 0.887 to 0.679. This was true for a few images, indicating that AutoGAD is perhaps more competitive with MPCA than originally thought. The effect of the $PA\ S\,NR$ threshold and slight nuances in MPCA are also evident in reducing the FPFs from Simple MPCA. Finally, it can be seen that $Y_{Initial} = 0.239$, which Jablonski used to help lower FPFs, greatly affects the TPFs rates in a few cases. These drops aren't quite as dramatic as they first appear due to the small number of target pixels, but they are of note nonetheless.

		AutoGAD (ICA)	AutoGAD (FA:0)	AutoGAD (FA:-Inf)	AutoGAD (FA:0, 20)	AutoGAD (FA Opt)	AutoGAD (FA Opt Abs)	MPCA (0.1)	MPCA (0.249)	MPCA (FA: 0.1)	MPCA (Simple)	MFA (Simple: 2k)
ARES1F	TPF	0.9841	0.9881	0.9881	0.9801	0.9682	0.9573	0.9891	0.9940	0.9930	0.9821	0.9940
	FPF	0.0389	0.0282	0.0386	0.0268	0.0367	0.0240	0.0259	0.0259	0.0291	0.0421	0.1384
ARES2F	TPF	0.9749	0.9772	0.9772	0.9772	0.9674	0.9446	0.9902	0.9837	0.9902	0.9902	0.9446
	FPF	0.0567	0.0391	0.0433	0.0658	0.0415	0.0344	0.0537	0.0767	0.0449	0.0874	0.0639
ARES3F	TPF	0.8272	0.8207	0.8207	0.9034	0.8552	0.8621	0.8414	0.8000	0.8000	0.8207	0.8069
	FPF	0.0623	0.0589	0.0589	0.0760	0.0415	0.0617	0.0266	0.0194	0.0221	0.0979	0.0883
ARES4F	TPF	0.7362	0.7706	0.7706	0.7706	0.7706	0.7706	0.6789	0.6789	0.6789	0.5872	0.5780
	FPF	0.0527	0.0302	0.0302	0.0461	0.0320	0.0230	0.0378	0.0285	0.0425	0.0482	0.0747
ARES1D	TPF	0.9850	0.7787	0.8851	0.8511	0.8681	0.7149	0.8979	0.7660	0.8638	0.5574	0.9319
	FPF	0.0450	0.0246	0.0556	0.0312	0.0518	0.0414	0.0313	0.0257	0.0276	0.0476	0.1044
ARES2D	TPF	0.9712	0.8987	0.9350	0.9541	0.9522	0.9369	0.9522	0.7591	0.9369	0.9159	0.9503
	FPF	0.0327	0.0436	0.0454	0.0345	0.0237	0.0094	0.0504	0.0361	0.0442	0.0331	0.0237
ARES1C	TPF	--	--	--	--	--	--	--	--	--	--	--
	FPF	0.0112	0.0000	0.0168	0.0000	0.0256	0.0183	0	0	0	0	0.2175
ARES2C	TPF	--	--	--	--	--	--	--	--	--	--	--
	FPF	0.0348	0.0275	0.0308	0.0253	0.0265	0.0260	0	0	0	0	0.2108
4Ships2	TPF	1	1	1	1	0.9910	0.9880	1	1	1	0	0.8163
	FPF	0.1758	0.1614	0.1829	0.2095	0.1353	0.1026	0.0178	0.0173	0.0167	0	0.1179
Virgin1	TPF	0.9875	0.9125	0.9125	0.9750	0.9500	0.9500	1	1	1	1	1
	FPF	0.1411	0.1315	0.1315	0.1223	0.0853	0.0576	0.1533	0.1180	0.1464	0.1511	0.1413

Table 6.4: Algorithm Comparison.

The remaining columns of Table 6.4 are variants of Algorithm 6.1, and MPCA with factors in some form instead of PCs. The *MPCA (FA:0.1)* column used a set of the rotated factors (MDSL criterion plus c_k retained) for D_1 and $Y_{Initial} = 0.1$. This provides no benefit, and could be partly due to the mix of factors and PCs in the D_i components. Not included in the table, but investigated, was rotating all of the PCs for a full-factor model within the MPCA framework. This lowered the TPF for ARES4F to 0.6422 and the TPF for

ARES1D to 0.7106, while other rates remained approximately the same. This could be due to the rotation incorporating too many factors and removing meaning of the factors, or simply because the parameters were not re-optimized, even though most of the rates remained similar. The final column used factors instead of PCs within Simple MPCA, where the trailing factors were the last k, and the major factors the first k, of a $2k$-rotated factor model. This generally increased the FPFs, but provided a better TPF on ARES1D. Using a full rotated factor model within the MPCA framework was also investigated, both in rotating the first k independently of the rest, and rotating all p factors simultaneously. Neither of these methodologies showed benefit.

The FA AutoGAD variants shown removed ICA entirely, as ICA on factors yields the same ICs as the PC case. Upon initial investigation, it became evident that the previous PA SNR and maximum score thresholds from Table 6.1 were too high for the factor scores. For example, the first factor for ARES1D that had such desirable properties and was previously shown in Figure 6.1(b), has a PA SNR below two and a maximum score below nine. Such occurrences suggested a certain homogeneity of the target factor scores. Further investigation revealed that PA SNR thresholds even as low as zero and $-\infty$ (equivalent to ignoring the threshold) still provided reasonable results across problems while only slightly increasing FPF. Reducing the maximum score threshold also helped, as did varying the IAN iterations, as shown by the *AutoGAD (FA:0,20)* column where $t_{SNR} = 0$ and I_h was lowered to 20 from the usual 100. This made sense in that too much filtering could smooth out targets where targets are more homogeneous, even though they are distinct from background. In these AutoGAD with FA results in Table 6.4, a maximum score threshold of 7 was used.

Clearly, for a factor-based technique the parameters required a new investigation and optimization, and the methodology required refinement. Part of the allure of MPCA is that it uses information from all PCs. Unfortunately, the D_i components are not directly

interpretable. FA also utilizes all information found in the image, in a way, by rotating the components to have high loadings. This is additionally beneficial in that the factors are then interpretable in terms of regions of the spectrum, and thus, materials in the image. Another consideration is the size of the factor model, and as to whether a full, rotated factor model is necessary. In practice, the desirable factors seemed somewhat invariant to small changes in c_k for retention, and the minor factors only seemed to duplicate information found in the primary factors or added noise. As anomalies contribute noise to the data, trailing PCs and factors may have anomalies contributing to them such that some background materials could score high. In this sense, it is not important to maintain any factors from outside a MDSL-criterion reduced model. Given the results from Table 6.4 and experimentation, and because a revised AutoGAD framework is highly reproducible, developing a new form of algorithm based on the AutoGAD framework was the best path forward. Admittedly, the FA-based MPCA algorithms were not optimized necessarily, but they are also problematic because the factor scores are combined and uninterpretable. Again, as on certain problems the factor scores are more homogeneous within-factor than they were in the PC case, this can further adversely affect the resulting D_i components for purposes of discrimination.

In order to improve AutoGAD, first, ICA is replaced with Varimax rotation factor analysis. This removes all randomness from the algorithm, assuming convergence of the rotation. This convergence was never an issue in practice. The factor analysis step also avoids the computational expense from ICA in not having to solve the statistical independence problem. There is still some expense due to the factor rotation, but it is very minimal. Next, IAN filtering is added immediately following the factor rotation, and before any factor thresholding. This takes place for $I_{initial}$ iterations between steps 12 and 13 in Algorithm 6.1. Such a technique showed promise in MPCA, and in experimentation helped to improve the PA SNR values on images and discriminatory factors where they were otherwise very low. Next, an initial set of RSM experiments were performed to

investigate the parameters and feasibility of this modified algorithm, as well as for purposes of developing a good operating point for further development. Experimentation in this manner was necessary as the interactions between parameters can be significant.

First, a 3^6 full-factorial design was run using the initial settings in Table 6.5 for seven images as a training set: ARES1F, ARES2F, ARES3F, ARES4F, ARES1D, ARES2D, and VirginIslands1. The ARES images were used due to their different characteristics and because they have truth masks, and the AVIRIS image was added to provide a different type of scene and sensor. c_k was not varied, as the specific eigenvalue cut-off did not affect the factors that greatly in initial experimentation. The low PA SNR threshold for smoothing was not varied or changed either here. $I_{initial}$ was set fairly low, keeping in mind those images like ARES1D where too much filtering could have an adverse effect on the scores. The two component threshold parameters were both varied over a range less than or equal to previous AutoGAD optimal settings as a result of the prior analysis and findings. The IAN filtering window size was not varied due to it being optimal in both AutoGAD and MPCA, and representing a desired smoothing sensitivity for the filter.

After these runs were complete, a second order response model with two-way interaction terms was constructed:

$$J(x) = \beta_0 + \sum_{i=1}^{s} \beta_i x_i + \sum_{i,j,i<j} \beta_{i,j} x_i x_j + \sum_{i=1}^{s} \beta_{i,i} x_i^2, \tag{6.5}$$

where s is the number of varied parameters. All terms except five of the interaction terms were statistically significant using $\alpha = 0.05$, and the resulting 23-term model had a $R^2_{adj} = 0.994$. Originally, to explore higher TPFs, the response and objective function to minimize was set to,

$$J(\mu_{TPF}, \mu_{FPF}, \sigma_{TPF}, \sigma_{FPF}) = 1 - \mu_{TPF} + \sigma_{TPF} + 2\mu_{FPF} + 2\sigma_{FPF}, \tag{6.6}$$

where these means and standard deviations were over the seven training images. This yielded optimal settings of approximately $t_{MS} = 6.19$, $t_{SNR} = -3$, $Y_{SNR} = Y_{ID} = 350$,

Table 6.5: Experiment 1 and 2 Settings.

Parameter	Name	Experiment 1	Experiment 2
c_k	MDSL Dimension Adjust	-1	-1
Y_{SNR}, Y_{ID}	Average Number Pixels/Bin	150, 300, 450,	200, 350, 500
t_{SNR}	PA SNR Threshold	-3, 0, 3	-3
t_{MS}	Max Score Threshold	5, 7, 9	6.19
t_l	Low PA SNR	10	10
I_h	IAN Filtering Iterations (High SNR)	0, 25, 50	31
I_l	IAN Filtering Iterations (Low SNR)	0, 10, 20	20
$I_{initial}$	IAN Filtering Iterations Initial	0, 2, 4	3
w	Window Size for IAN Filter	3	3

$I_h = 31$, $I_l = 20$, and $I_{initial} = 3$. Beyond this optimization, given that original AutoGAD yielded such low FPF rates when using different bin widths for component selection and anomaly declaration, the idea to allow $Y_{ID} \neq Y_{SNR}$ needed to be explored. Again, a three level design and resulting model were used, but now only the two bin parameters were varied. These were centered at the optimal just found. These settings are shown in Table 6.5 as Experiment 2. For this second test, a response similar to that used by Jablonski was used to emphasize lower FPFs, with an additional term added to provide more consistent TPFs:

$$J(\mu_{TPF}, \mu_{FPF}, \sigma_{TPF}, \sigma_{FPF}) = (1 - \mu_{TPF})^2 + \sigma_{TPF}^2 + 3\mu_{FPF}^2 + 3\sigma_{FPF}^2. \tag{6.7}$$

Only one term in the resulting model was statistically insignificant, yielding a model with $R_{adj}^2 = 0.999$, and the settings $Y_{SNR} = Y_{ID} = 500$. This was somewhat surprising as it may have been expected that this would greatly increase the FPF rate. In fact, this gave lower FPFs on the AVIRIS data. The specific results using these new optimal settings for several

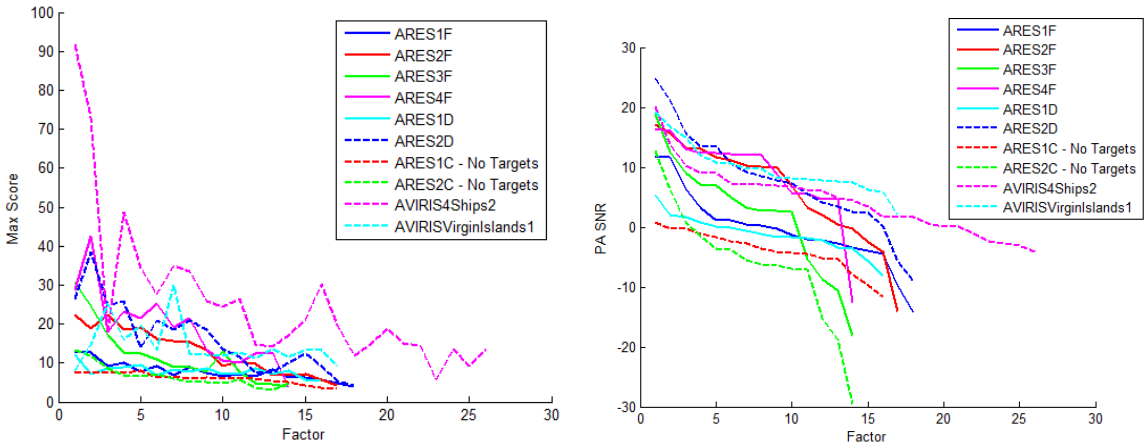
problems were already shown in Table 6.4 in the *AutoGAD (FA Opt)* column. These results showed great promise in providing better rates in some cases than MPCA or the original AutoGAD algorithm. For the results in the *AutoGAD (FA Opt Abs)* column of Table 6.4, one additional adjustment was made to the methodology. Here, the absolute value of each factor mapping was taken before any histogram construction. This aided FPF rates, but affected the TPF in a significant manner for ARES1D. Using the absolute value, only one side of a score map had to be checked for potential anomalies with a large magnitude while still considering the scores on both sides of the original score map. However, this also reduced TPFs because more variation was added to the background estimates in the histograms.

6.4 Investigating Specifics of the Framework

The experiments done in the previous section revealed a decent setting and algorithm for anomaly detection using factor analysis. However, they also suggested that a more rigorous approach and further deviation from the standard AutoGAD framework could provide more improvement. Specifically, that rules adapting to the images and factors would yield better results for certain images such as ARES1D, ARES1C, ARES2C, and ARES4F.

Trying to develop a way or rules to identify images and/or factors that require special consideration proved very difficult. Figure 6.6(a) shows the maximum factor scores, sorted by magnitude, for the set of 10 images. Likewise, Figure 6.6(b) shows the PA SNR for those factors in the same order.

From these plots, it can be seen that ARES1D, ARES1C, and ARES2C are significantly more homogeneous than the other images. This is to be expected for ARES1C and ARES2C as they have no anomalies. Meanwhile, the AVIRIS images have a generally higher PA SNR and maximum scores, especially so for Scene1 and Ship1,



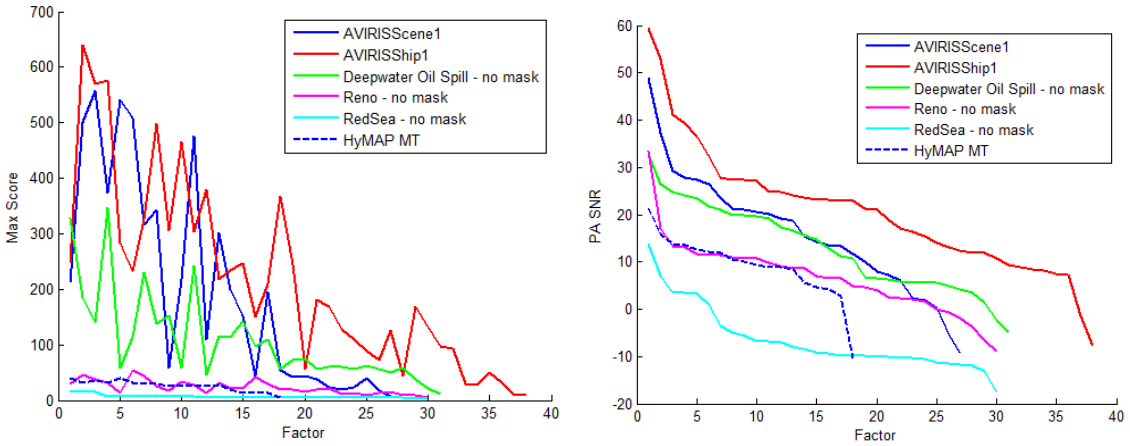
(a) Max Scores.

(b) PA SNR.

Figure 6.6: Training Set Max Scores and PA SNRs.

as shown in Figure 6.7. From the experiments so far, it has been shown that changing parameters to increase TPF on ARES1D and others also increases FPF on ARES1D, ARES1C, and ARES2C. The interactions amongst parameters makes optimizing all images simultaneously much more complex.

FPF rates are generally higher on the AVIRIS images, and so similarly it could be desirable to detect and distinguish such data sets. Interestingly, MPCA had TPFs of 0.586 and 0.321, and FPFs of 0.001 and 0.003 on the Scene1 and Ship1 images, while the thus far optimal factor analysis algorithm had TPFs of 0.9977 and 0.9844, and FPFs of 0.0584 and 0.1033, respectively. The factor analysis appears to be somewhat better at detecting full-pixel targets, while MPCA is somewhat better at reducing false positives. Now, consider the PA SNRs after initial IAN filtering for the training set of images, shown in Figure 6.8. The PA SNRs of ARES1C, ARES2C, and ARES1D improve after the filtering. This is beneficial for ARES1D, but not images ARES1C and ARES2C that have no targets, as this leads to false positives by making certain background pixels further separated.



(a) Max Scores. (b) PA SNR.

Figure 6.7: Other Images' Max Scores and PA SNRs.

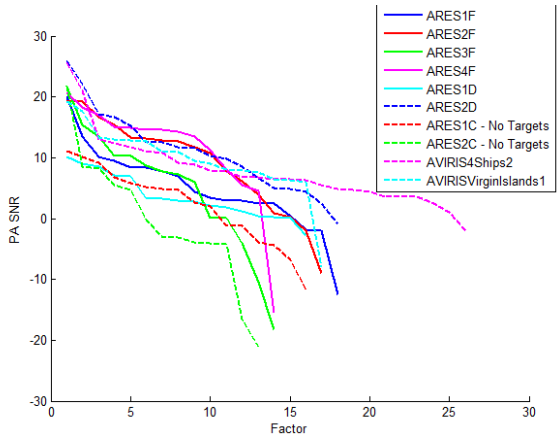


Figure 6.8: Training PA SNRs After $I_{init} = 3$.

In order to try and develop a set of rules to detect characteristics of an image and adjust parameters accordingly for a more robust algorithm, the seven-image training set was split into two subsets based on mean PA SNR. ARES1D, ARES1C, and ARES2C have

the lowest mean PA SNRs for the retained factors. Meanwhile, the AVIRIS images have the highest. These are shown for the training set in Table 6.6.

Table 6.6: Mean PA SNRs.

Image	Mean Pre-IAN	Mean Post-Initial IAN
ARES1F	-0.4641	4.8604
ARES2F	6.0941	8.9990
ARES3F	1.9905	4.8934
ARES4F	8.7720	11.1967
ARES1D	-1.2688	3.5736
ARES2D	7.5211	10.9008
ARES1C	-4.0207	2.0793
ARES2C	-6.0631	-0.5746
4Ships2	4.4274	7.8261
VirginIslands1	9.7433	9.6085

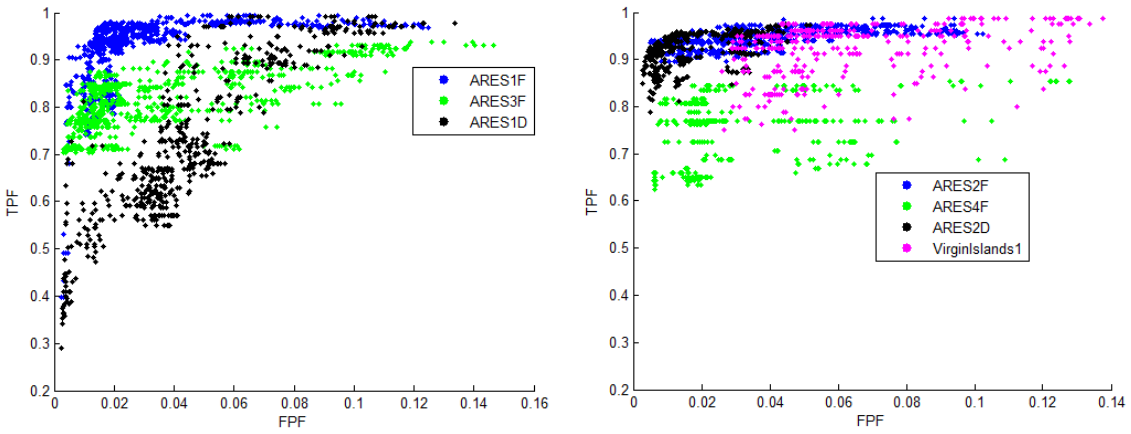
It seems that images with factors that likely need to be smoothed less have generally lower PA SNRs, and images with factors that likely need to be smoothed more have generally higher PA SNRs. Additionally, this mean metric is more telling after the initial IAN filtering has been applied. Due to having the lowest means, ARES1F, ARES3F, and ARES1D were chosen as one subset (from the seven-image training set used previously). A 3^7 full-factorial design was done over the parameters and settings in Table 6.7, performed over wide ranges on the seven test images to try and learn over-arching trends. Again a second-order model was fit, evaluated, and optimized using Equation 6.7 as a response.

Table 6.7: Experiment 3 and Optimal Settings.

Parameter	Name	Settings	Optimal
c_k	MDSL Dimension Adjust	-1	-1
Y_{SNR} Y_{ID}	Average Number Pixels/Bin	200, 450, 700,	241
t_{SNR}	PA SNR Threshold	-27, -12, 3	-27
t_{MS}	Max Score Threshold	2, 7, 12	5.5
t_l	Low PA SNR	0, 7.5, 15	15
I_h	IAN Filtering Iterations (High SNR)	10, 55, 100	58
I_l	IAN Filtering Iterations (Low SNR)	5, 20, 35	12
$I_{initial}$	IAN Filtering Iterations Initial	0, 3, 6	0
w	Window Size for IAN Filter	3	3

However, this response, model, and optimization was done three ways. First, using all seven images, then using ARES1F, 3F, and 1D as a training set *Low*, and then using the remaining four images as a training set *High*. This was done in this manner to investigate if different sets of parameters could enable flexibility towards those images with lower PA SNRs, assuming some criterion such as mean PA SNR could be used to choose a set of parameters. Figure 6.9 depicts the TPF and FPF rates at each design point for the seven images.

The optimal design point in all cases did not yield consistently improved rates across images for this experiment number 3, as depicted in Table 6.8, despite models with $R^2_{adj} > 0.97$. Here, a comparison is given to AutoGAD, MPCA with $Y_{initial} = 0.1$ due to its increased TPFs, and the previous optimal settings (*i.e.*, Table 6.4). The seven-image optimal improved some TPFs with minimal impact to corresponding FPFs, but a few images had significant drops in TPFs or increases in FPFs. The three-image optimal



(a) Design Points Set 1.

(b) Design Points Set 2.

Figure 6.9: Experiment 3 Rates.

did improve those three images somewhat, but not necessarily to a significant effect. On the contrary, the four-image optimal vastly decreased false positives, but at the expense of true positives. This suggested that a decision rule for an entire image on the PA SNR may not be adaptive enough to improve results.

The effect of smoothing was clear at this point, and so for a fourth experiment the smoothing iterations were set equal to one another. In order to further investigate whether the algorithm should adapt to an image as a whole, or its factors, the experiment with settings and subsequent optimal as shown in Table 6.9 was performed. Here, the bin widths were allowed to vary according to being above (Y_{high}) or below (Y_{low}) the PA SNR threshold t_l. A model was fit and optimized for all three sets of images again, where the design responses were as shown in Figure 6.10. Previous models had shown that interactions between most of the parameters are significant, and this was evident in the results here. All parameters were significant by themselves or as part of an interaction, with R^2_{adj} values over 0.98 for the three models with statistically significant terms included. In this case,

						Experiment 3			Experiment 4		
		AutoGAD	MPCA (0.1)	Opt1 No Abs	Opt1 Abs	All	Low	High	All	Low	High
ARES1F	TPF	0.9841	0.9891	0.9682	0.9573	0.9772	0.9791	0.8371	0.9126	0.8928	0.8371
	FPF	0.0389	0.0259	0.0367	0.0240	0.0320	0.0411	0.0054	0.0133	0.0166	0.0159
ARES2F	TPF	0.9749	0.9902	0.9674	0.9446	0.9511	0.9674	0.9446	0.8990	0.8958	0.8893
	FPF	0.0567	0.0537	0.0415	0.0344	0.0494	0.0565	0.0161	0.0115	0.0229	0.0291
ARES3F	TPF	0.8272	0.8414	0.8552	0.8621	0.8138	0.8069	0.7517	0.7517	0.7655	0.7724
	FPF	0.0623	0.0266	0.0415	0.0617	0.0603	0.0583	0.0097	0.0148	0.0299	0.0178
ARES4F	TPF	0.7362	0.6789	0.7706	0.7706	0.8257	0.7248	0.8165	0.5780	0.5780	0.6330
	FPF	0.0527	0.0378	0.0320	0.0230	0.0343	0.0342	0.0185	0.0212	0.0187	0.0192
ARES1D	TPF	0.9850	0.8979	0.8681	0.7149	0.8766	0.9064	0.3830	0.6681	0.6383	0.6298
	FPF	0.0450	0.0313	0.0518	0.0414	0.0566	0.0642	0.0036	0.0285	0.0289	0.0358
ARES2D	TPF	0.9712	0.9522	0.9522	0.9369	0.9541	0.9159	0.9254	0.6998	0.6979	0.7380
	FPF	0.0327	0.0504	0.0237	0.0094	0.0244	0.0305	0.0059	0.0130	0.0125	0.0115
ARES1C	TPF	--	--	--	--	--	--	--	--	--	--
	FPF	0.0112	0	0.0256	0.0183	0.0375	0.0411	0	0.0083	0.01628	0.0169
ARES2C	TPF	--	--	--	--	--	--	--	--	--	--
	FPF	0.0348	0	0.0265	0.0260	0.0400	0.0486	0.0030	0.0083	0.0129	0.0127
4Ships2	TPF	1	1	0.9910	0.9880	0.9970	1.0000	0.9849	1	1	0.9970
	FPF	0.1758	0.0178	0.1353	0.1026	0.1708	0.2180	0.0829	0.0825	0.0727	0.0785
Virgin1	TPF	0.9875	1	0.9500	0.9500	0.9500	0.8875	0.9500	0.7750	0.7750	0.7875
	FPF	0.1411	0.1533	0.0853	0.0576	0.1124	0.1251	0.0428	0.0417	0.0419	0.0430

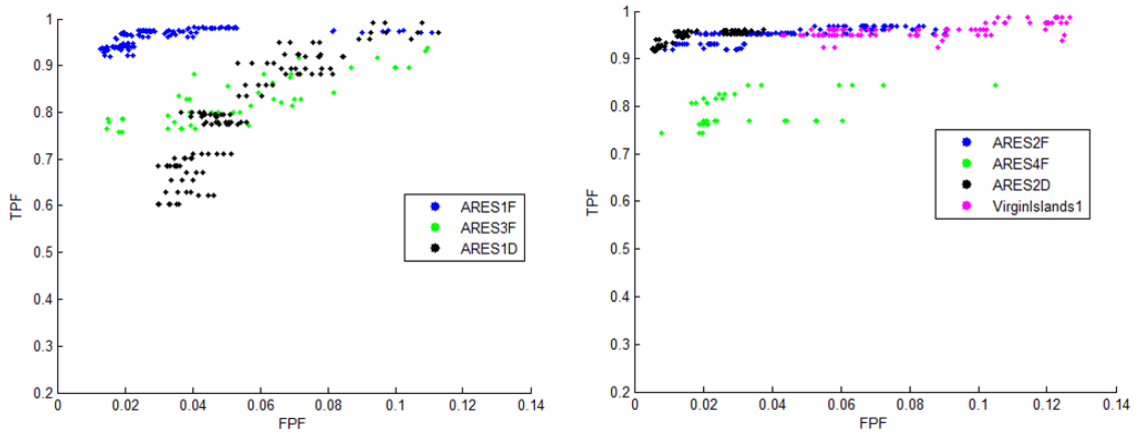
Table 6.8: Experiment Comparison.

the settings were in no way ideal. Primarily, these experiments indicated that adjusting parameters on characteristics of the entire image may not be sensitive enough, rather, that it is more important to consider the individual factors. Aside from score homogeneity influencing PA SNR, additional concepts were also affecting the algorithm performance.

Table 6.9: Experiment 4 and Optimal Settings.

Parameter	Name	Settings	Optimal
c_k	MDSL Dimension Adjust	-1	-1
Y_{low}	Average Number Pixels/Bin	200, 400, 600,	200
Y_{high}	Average Number Pixels/Bin	200, 400, 600,	600
t_{SNR}	PA SNR Threshold	-27, -15, -3	-15.25
t_{MS}	Max Score Threshold	5, 6, 7	7
t_l	Low PA SNR	15	15
$I_h = I_l$	IAN Filtering Iterations	10, 35, 60	60
$I_{initial}$	IAN Filtering Iterations Initial	0, 3, 6	5
w	Window Size for IAN Filter	3	3

Consider Algorithm 6.1 again. In step 12, the maximum score was taken before filtering. After quick investigation, it was confirmed that this should take place after the initial filtering. The first zero-bin histogram also has an interesting sensitivity when determining PA SNR. Figure 6.11 shows an example histogram of scores from ARES1F. The first zero-bin histogram method finds the first zero count bin to the right of the center of the scores, and uses this to separate the pixels into potential anomalies and background. Therefore, if a component has different sets of anomalies, one set can alter the background



(a) Design Points Set 1.

(b) Design Points Set 2.

Figure 6.10: Experiment 4 Rates.

variance and make it more difficult to detect a component with true high SNR. Alternatively, a first zero-bin on each side of the center could be found, where the scores in between could be used to build the background variance. Although this latter "double-zero" method did positively affect PA SNR estimates for a few components across images, in practice, it was not significant and in some cases it also served to increase false positives. For those factors with multiple classes of targets, there was typically another factor on which one of the sets was also anomalous. Figure 6.11 is a good reference to show how bin width size can affect the histogram method. Larger widths move the first zero to the right, this decreasing FPF and TPF, while smaller widths move it to the left, increasing TPF and FPF. Given a decision rule, this could be the best way to dynamically adjust the over-arching algorithm to characteristics of the factor. In other words, factors with well-separated anomalies and high PA SNR can support larger bin widths, while those with lower PA SNR that are not well separated require a smaller bin width. Therefore, intelligent choices of smoothing iterations

and bin width for each individual factor can appropriately adjust the scores and sensitivity of the histogram, resulting in easier detection of anomalies and less false positives.

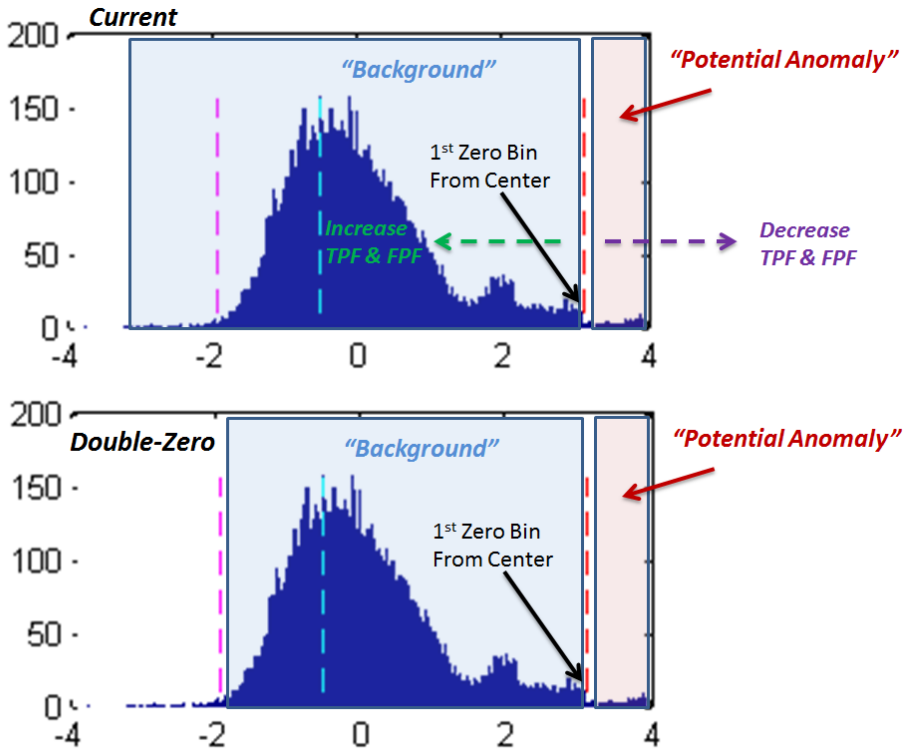
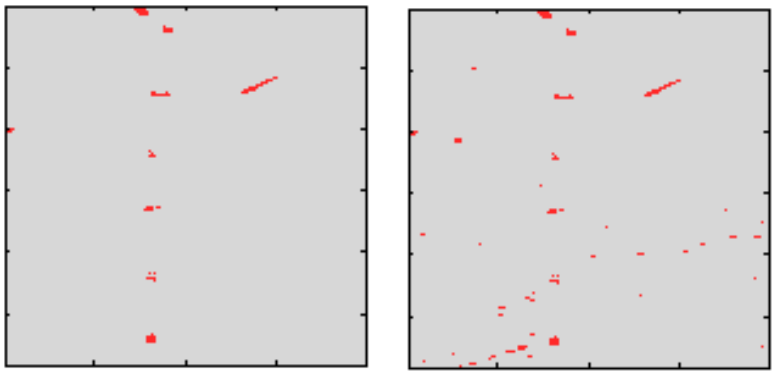


Figure 6.11: Zero-Bin Considerations.

Other factors not yet discussed also affect algorithm performance: anomalies in the covariance estimate, thresholding both sides of the scores, using a fixed bin width for each histogram, and sensor error present in pixels. Table 6.10 shows results for variations involving some of these aspects and some already discussed, where the optimal from Experiment 2 (*Opt1*) was used as a basis. Using a double-zero histogram (*DBZH*) for a better background variance estimate provided little benefit over the improvement shown by just thresholding the positive side of the factor scores (*1 Side*). Iteration, by removing potential anomalies with a score $\geq 2.5 \times t_{MS}$ on any retained factor, helped to reduce false

positives. Doing so repeatedly however, waiting for convergence of anomalies (*Iter Conv*), did not prove beneficial and yielded factors that were too clean such that more background was erroneously identified as anomalous. Eliminating all of the IAN filtering, except for the initial smoothing, yielded exceptional TPF rates for ARES1D and ARES4F, making it obvious that smoothing on those images can actually make it more difficult to identify anomalies. Such a case is shown in Figure 6.12, where potential anomalies are shown before and after a low number of iterations of IAN filtering for the ARES1D image. Using a variable bin width without scaling according to the number of pixels, *i.e.*, Equation 6.3 times N, with $Y = 0.0008 \times N$ to be near $Y = 500$, provided benefit in some cases (*Var Bin*). However, given the greater homogeneity of factor scores for images such as ARES1D, it also causes issues in some cases, where fitting to the range of scores makes the histograms too sensitive or not sensitive enough. This again points towards the need to develop a decision rule that increases sensitivity for certain images, and decreases sensitivity for others. Jablonski's original variable bin width, Equation 6.3, is also investigated shortly.



(a) Original. (b) After Smoothing.

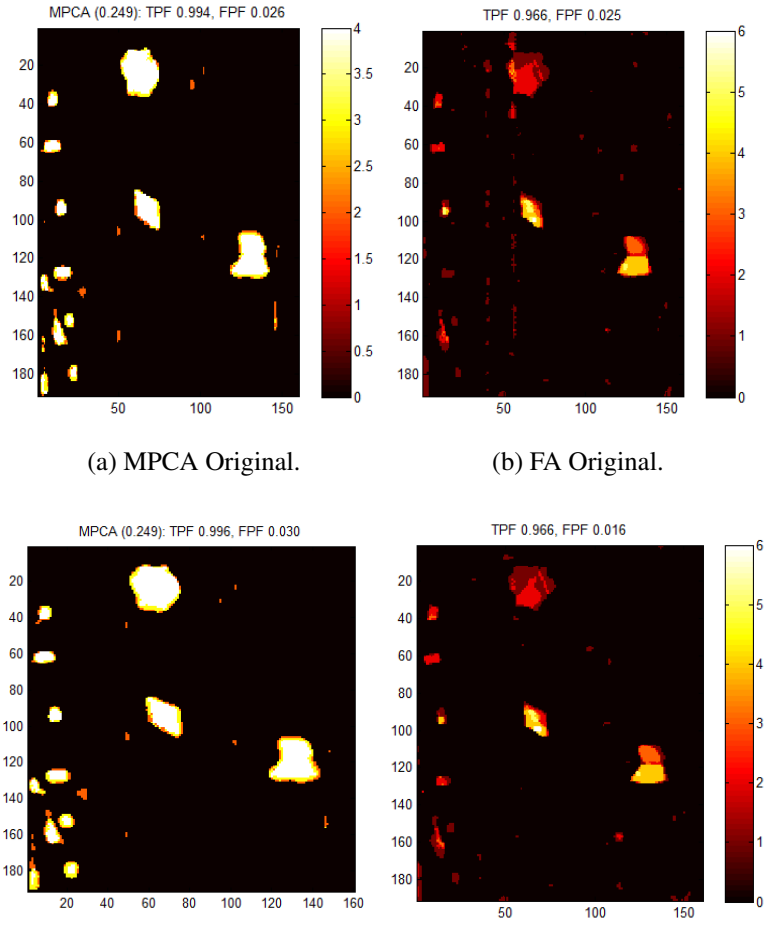
Figure 6.12: Potential Anomalies.

					Bin Width = Y/N=500/N							
		AutoGAD	MPCA (0.249)	MPCA (0.1)	Opt1	1 Side, DBZH	2 Sides, 1 Iter	1 Side, 1 Iter	1 Side, Iter Conv	1 Side, 1 Iter, DBZH	1 Side, 1 Iter, No Smooth	1 Side, 1 Iter, Var Bin
ARES1F	TPF	0.9841	0.9940	0.9891	0.9662	0.9603	0.9662	0.9573	0.9573	0.9603	0.9652	0.9940
	FPF	0.0389	0.0259	0.0259	0.0246	0.0154	0.0246	0.0139	0.0139	0.0154	0.0756	0.0282
ARES2F	TPF	0.9749	0.9837	0.9902	0.9674	0.9674	0.9805	0.9707	0.9739	0.9707	0.9577	0.9479
	FPF	0.0567	0.0767	0.0537	0.0415	0.0364	0.0434	0.0302	0.0225	0.0302	0.0782	0.0283
ARES3F	TPF	0.8272	0.8000	0.8414	0.8552	0.8345	0.8621	0.8414	0.7724	0.8414	0.8483	0.8552
	FPF	0.0623	0.0194	0.0266	0.0415	0.0370	0.0478	0.0391	0.0286	0.0391	0.1122	0.0680
ARES4F	TPF	0.7362	0.6789	0.6789	0.7706	0.7706	0.7706	0.7706	0.7706	0.7706	0.8349	0.7706
	FPF	0.0527	0.0285	0.0378	0.0320	0.0277	0.0293	0.0262	0.0241	0.0262	0.0545	0.0609
ARES1D	TPF	0.9850	0.7660	0.8979	0.8681	0.8128	0.8681	0.8128	0.8128	0.8128	0.9915	0.7872
	FPF	0.0450	0.0257	0.0313	0.0518	0.0415	0.0518	0.0415	0.0415	0.0415	0.1484	0.0451
ARES2D	TPF	0.9712	0.7591	0.9522	0.9522	0.7935	0.8776	0.8700	0.8184	0.8700	0.9618	0.8910
	FPF	0.0327	0.0361	0.0504	0.0237	0.0126	0.0272	0.0162	0.0171	0.0162	0.0223	0.0163
ARES1C	TPF	--	--	--	--	--	--	--	--	--	--	--
	FPF	0.0112	0	0	0.0133	0.0094	0.0133	0.0094	0.0094	0.0094	0.0486	0.0348
ARES2C	TPF	--	--	--	--	--	--	--	--	--	--	--
	FPF	0.0348	0	0	0.0246	0.0174	0.0246	0.0174	0.0174	0.0174	0.0295	0.0197
4Ships2	TPF	1	1	1	0.9910	0.9910	1	1	1	1	0.9940	0.9608
	FPF	0.1758	0.0173	0.0178	0.1353	0.0906	0.1358	0.0891	0.0923	0.0911	0.2183	0.0054
Virgin1	TPF	0.9875	1	1	0.9500	0.9250	0.9750	0.9500	0.9500	0.9500	0.9750	0.9500
	FPF	0.1411	0.1180	0.1533	0.0853	0.0738	0.0806	0.0678	0.0678	0.0671	0.0811	0.1277

Table 6.10: Techniques Investigation Results.

The bands kept in Chapter 4 included a few where the sensor yielded an artifact or line of erroneous pixels and where the remainder of the pixels in the band are not noisy. A few of these bands were also retained by Smetek [191] and Johnson [110] in their band selection analysis. Interestingly, AutoGAD and MPCA are not affected by the presence of these bands. However, these pixels present themselves as anomalous on any factor that has those bands highly weighted. For example, the final score maps for MPCA and the *Opt1* FA framework are shown in Figure 6.13 for ARES1F, where the heat map number represents the number of components on which that pixel was declared anomalous. As it turns out, a large number of false positives for the FA algorithms were due to this phenomena, and removal of those three or four bands reduced false positives in many cases for the HYDICE data. Identification of these bands is very easy to do visually, should such artifacts be present in a hyperspectral image being analyzed with

a FA method. Alternatively, experimentation showed that a slightly different framework and better understanding of the interactions between parameters could also help to reduce the false positives.



(a) MPCA Original. (b) FA Original.

(c) MPCA Fewer Bands. (d) FA Fewer Bands.

Figure 6.13: Comparison With/Without Sensor Error.

6.5 Global Factor Analysis-Based Anomaly Detector (GFAAD)

The previous findings ultimately lead to an improved methodology, referred to as the Global Factor Analysis Anomaly Detector (GFAAD). The framework is shown in

Figure 6.14, and the algorithm is shown in full as Algorithm 6.2. The primary change to methodology was the idea to adapt the bin width and the smoothing iterations for each factor based on its scores' PA SNR. Note, a PA SNR threshold was no longer used to nominate factors for consideration either. This was designed to aid detection in those images with more homogeneous pixels.

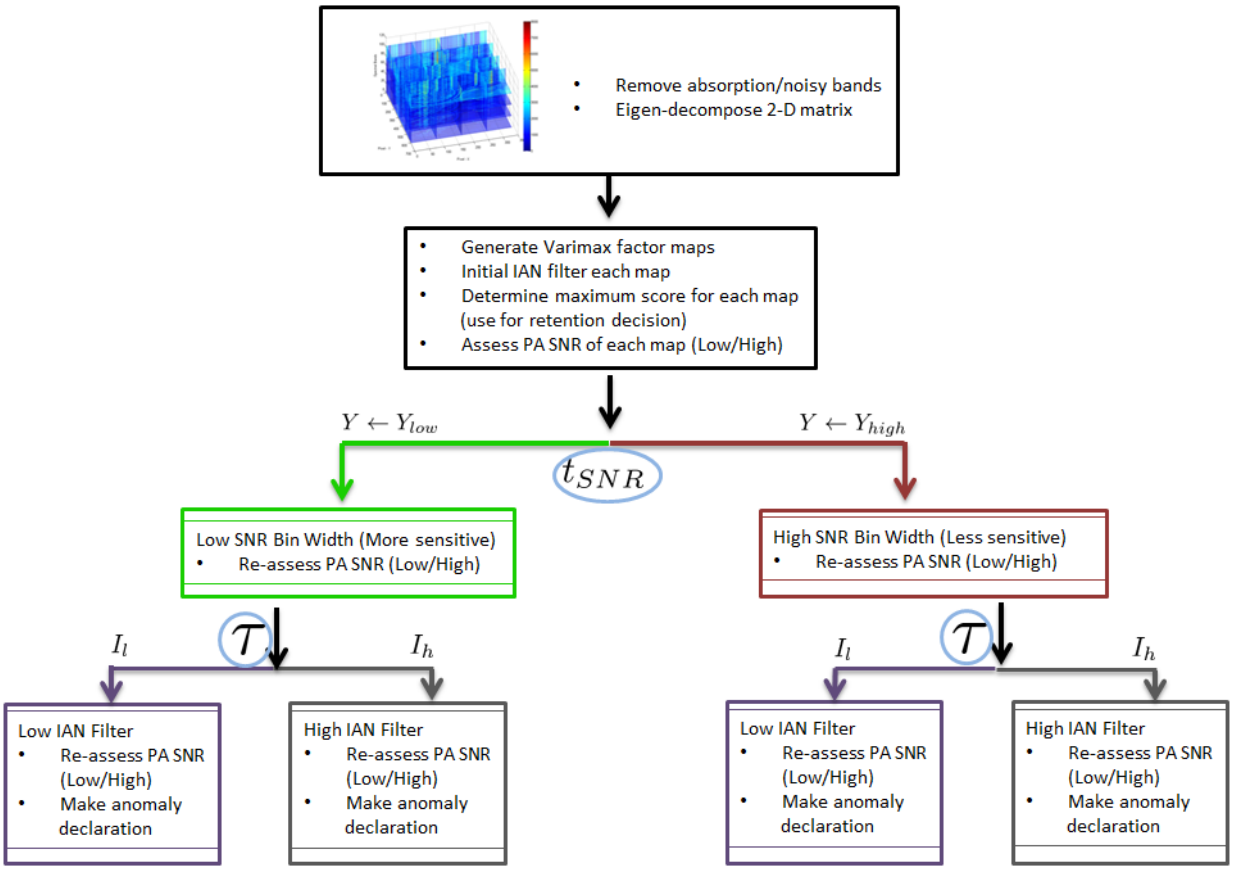


Figure 6.14: General GFAAD Process.

Again, this algorithm shares many characteristics with AutoGAD and MPCA. However, ICA and its expense and variability have been removed, an initial filtering has been added, and decision steps have been added to adapt to the image. In MPCA, Jablonski

Algorithm 6.2 GFAAD Algorithm

1: $X^c_{N\times p} \leftarrow \left(X_{N\times p} - \mathbf{1}_{N\times 1}\boldsymbol{\mu}^T\right)$: data is centered.

2: Find eigenvectors V and eigenvalues Λ from $cov\left(X^c_{N\times p}\right)$: do PCA.

3: $k \leftarrow MDSL(\Lambda)$. $L_{p\times k} \leftarrow V_{p\times k}\Lambda^{1/2}$ denotes the factor loadings. Varimax rotate the loadings to yield $\hat{L}_{p\times k}$. Compute the factor scores: $F_{N\times k} \leftarrow X^c_{N\times p}\hat{L}_{p\times k}\left(\hat{L}^T_{p\times k}\hat{L}_{p\times k}\right)^{-1}$.

4: **if** $|\min(F^i)| > \max(F^i)$ for $1 \le i \le k$ **then**

5: $\quad F^i \leftarrow -F^i$.

6: **end if**

7: $F^i \leftarrow IAN(F^i)$, with $I_{initial}$ iterations. For each factor mapping F^i, $m_i \leftarrow \max(F^i)$.

8: Retain any factor mapping with $m_i \ge t_{MS}$.

9: $snr_i \leftarrow PA\ SNR(F^i)$ with bin width $Y_{Initial}/N$ using first zero-bin histogram.

10: **if** $snr_i \le t_{SNR}$ **then**

11: $\quad Y^i \leftarrow Y_{low}$. $snr_i \leftarrow PA\ SNR(F^i)$ with bin width Y_{low}/N.

12: **else**

13: $\quad Y^i \leftarrow Y_{high}$. $snr_i \leftarrow PA\ SNR(F^i)$ with bin width Y_{high}/N.

14: **end if**

15: **if** $snr_i \le \tau$ **then**

16: $\quad F^i \leftarrow IAN(F^i)$, with I_h iterations.

17: **else**

18: $\quad F^i \leftarrow IAN(F^i)$, with I_l iterations.

19: **end if**

20: Repeat Steps 9-14 using Y^i as the initial bin width.

21: Define $\eta_i \leftarrow PA\ SNR(F^i)$ as the threshold from the first zero-bin histogram, using Y^i.

22: If first iteration, remove any pixel j with $F^i_j > 2.5 \times t_{MS}$ from data used for covariance estimate and go to Step 2. Otherwise, proceed.

23: **if** $F^i_j > \eta_i$ **then**

24: $\quad$ Declare pixel j anomalous.

25: **end if**

added steps to ignore components if SNR was too low, but here, the algorithm adapts before assessing factors.

In order to assess this algorithm, first, a 3^6 full factorial design was used to vary parameters in the case of no IAN filtering after the initial smoothing. This lack of filtering also saves computational expense. Parameter settings are shown in Table 6.11 for this experiment as Experiment 5. All parameters were significant directly or in an interaction in a second-order model with interaction terms, with the exception of $Y_{initial}$. This makes sense, as this parameter is used mainly to determine how to adapt to each factor. R^2_{adj} of the resulting model was 0.9889. Optimization of Equation 6.7 yielded an optimal of $t_{MS} = 7.05$, $t_{SNR} = 15$, $I_{initial} = 5$, $Y_{initial} = 500 = Y_{low}$, and $Y_{high} = 650$.

Table 6.11: GFAAD Experiment Settings.

Parameter	Name	Experiment 5	Experiment 6
t_{MS}	Max Score Threshold	4, 6.5, 9	7.05
t_{SNR}	Bin Width PA SNR Threshold	0, 7.5, 15	5, 12.5, 20
$I_{initial}$	Initial IAN Iterations	0, 3, 6	5
I_h	IAN Iterations High	0	0, 30, 60
I_l	IAN Iterations Low	0	0, 15, 30
$Y_{initial}$	Pixels Per Bin Initial	400, 500, 600	500
Y_{low}	Pixels Per Bin Low	200, 350, 500	200, 350, 500
Y_{high}	Pixels Per Bin High	500, 650, 800	400, 550, 700
τ	Smoothing PA SNR Threshold	0	-10, 0, 10

Results on the seven image training set and six other images are shown in Table 6.12. Experiment 6 used this optimal as a basis, and varied the filtering iterations and bin sizes per the settings shown in Table 6.11. In this case, every varied parameter had an associated

statistically significant term, and the optimized response for Equation 6.7 yielded settings of $I_h = 40$, $I_l = 0$, $Y_{low} = 381$, $Y_{high} = 546$, and $\tau = 3$. The resulting TPF and FPF rates are highly competitive with MPCA and AutoGAD, while being more computationally efficient. Additionally, AutoGAD ran into memory issues during the ICA step on very large problems such as AVIRIS Scene1, while MPCA did not perform well on those problems. Here, the algorithms were being run on an Intel^{TM} Core i7 CPU Q840@1.87 GHz, 64-bit OS, with 8 GB RAM. Times shown for AutoGAD are an average of 20 runs, or represent the point at which memory was deemed too full.

GFAAD shows a flexibility to work well on a large variety of images with common settings. Yet, when analyzing the factor maps, it seemed false positives could still be made lower for certain images, and true positives in problems such as ARES1D could be made higher. Investigation showed that the interaction of IAN smoothing and bin size were critical to these issues. Additionally, it became clear that certain low PA SNR factors inflated false positives after smoothing, as the filtering generated false positives. Using a modified bin size equation, $Y \times (\max(scores) - \min(scores))$ also helped to reduce false positives on a few problems, to include yielding a TPF of 1 and FPF of 0.0034 in 4Ships2. Unfortunately, this bin size yielded poor results on many of the other AVIRIS images; for example, a less than 0.1 TPF on Ship1. It was clear that given some added complexity to Algorithm 6.2, in general, FPFs could be reduced and problems with low TPFs such as ARES1D could be improved. To do this, better consideration for PA SNR and smoothing were added into the algorithm. Figure 6.15 depicts the new, resulting framework and Algorithm 6.3 shows the algorithm in full. This refined algorithm is referred to as the Improved Global Factor Analysis Anomaly Detector (IGFAAD).

In IGFAAD, a few considerations are added. Step 6 is added before the initial IAN filtering in order to remove very low SNR factors that can become prone to false positives after smoothing. This also can aid in efficiency as fewer factors are being smoothed. The

		AutoGAD	MPCA (0.1)	MPCA (0.249)	GFAAD Exp 5	GFAAD Exp 6
ARES1F	TPF	0.9841	0.9891	0.9940	0.9662	0.9643
	FPF	0.0389	0.0259	0.0259	0.0154	0.0216
	Time (s)	1.1954	3.4239	3.3048	0.4996	0.6242
ARES2F	TPF	0.9749	0.9902	0.9837	0.9414	0.9609
	FPF	0.0567	0.0537	0.0767	0.0201	0.0231
	Time (s)	2.1267	4.8439	4.1189	1.5013	1.3217
ARES3F	TPF	0.8272	0.8414	0.8000	0.8414	0.8414
	FPF	0.0623	0.0266	0.0194	0.0584	0.0562
	Time (s)	6.2037	3.4645	3.3309	0.9667	1.3289
ARES4F	TPF	0.7362	0.6789	0.6789	0.8440	0.8532
	FPF	0.0527	0.0378	0.0285	0.0309	0.0308
	Time (s)	4.3283	1.9803	1.7072	0.6231	0.6216
ARES1D	TPF	0.9850	0.8979	0.7660	0.7489	0.8383
	FPF	0.0450	0.0313	0.0257	0.0260	0.0267
	Time (s)	2.3988	5.9893	5.7221	0.8016	1.3970
ARES2D	TPF	0.9712	0.9522	0.7591	0.9388	0.9426
	FPF	0.0327	0.0504	0.0361	0.0154	0.0157
	Time (s)	7.5703	2.9470	2.7712	0.8296	0.8777
ARES1C	TPF	--	--	--	--	--
	FPF	0.0112	0	0	0.0042	0.0082
	Time (s)	0.6396	2.7288	2.5232	0.3352	0.4263
ARES2C	TPF	--	--	--	--	--
	FPF	0.0348	0	0	0.0129	0.0145
	Time (s)	1.1739	2.7494	2.6025	0.3210	0.3305
4Ships2	TPF	1	1	1	0.9940	0.9940
	FPF	0.1758	0.0178	0.0173	0.1032	0.1020
	Time (s)	32.5945	35.4932	33.9093	17.5601	20.1354
Virgin1	TPF	0.9875	1	1	0.9875	0.9750
	FPF	0.1411	0.1533	0.1180	0.0947	0.0934
	Time (s)	1.1966	2.4670	2.3581	0.8268	0.8253
Scene1	TPF	--	0.6000	0.5862	0.9887	0.9887
	FPF	--	0.0009	0.0010	0.0375	0.0426
	Time (s)	38.4585	55.8701	54.6015	33.6119	36.1333
Ship1	TPF	--	0.4029	0.3210	0.9229	0.9483
	FPF	--	0.0034	0.0029	0.0668	0.0704
	Time (s)	50.1798	36.8604	35.0338	26.5833	31.1026
HyMAP	TPF	--	0.4483	0.4000	0.7103	0.5793
	FPF	--	0.0829	0.0814	0.0853	0.0847
	Time (s)	10.3844	14.4894	13.7981	7.6465	7.7123

Table 6.12: GFAAD Optimization Results.

Algorithm 6.3 IGFAAD Algorithm

1: $X^c_{N\times p} \leftarrow \left(X_{N\times p} - \mathbf{1}_{N\times 1}\boldsymbol{\mu}^T\right)$: data is centered.

2: Find eigenvectors V and eigenvalues Λ from $cov\left(X^c_{N\times p}\right)$: do PCA.

3: $k \leftarrow MDSL(\Lambda)$. $L_{p\times k} \leftarrow V_{p\times k}\Lambda^{1/2}$ denotes the factor loadings. Varimax rotate the loadings to yield $\hat{L}_{p\times k}$. Compute the factor scores: $F_{N\times k} \leftarrow X^c_{N\times p}\hat{L}_{p\times k}\left(\hat{L}^T_{p\times k}\hat{L}_{p\times k}\right)^{-1}$.

4: **if** $|\min(F^i)| > \max(F^i)$ for $1 \le i \le k$, **then** $F^i \leftarrow -F^i$, **end if**.

5: $snr_i \leftarrow PA\ SNR(F^i)$ with bin width $Y_{initial}/N$ using first zero-bin histogram.

6: **if** $snr_i > t_{SNR}$, **then** retain F^i, **end if**.

7: $F^i \leftarrow IAN(F^i)$, with $I_{initial}$ iterations. For each factor mapping F^i, $m_i \leftarrow \max(F^i)$.

8: Retain any factor mapping with $m_i \ge t_{MS}$. If none satisfy this, declare no anomalies and stop. Otherwise, go to Step 9.

9: $snr_i \leftarrow PA\ SNR(F^i)$ with bin width $Y_{initial}/N$ using first zero-bin histogram.

10: **if** $snr_i \le \tau_1$, **then** $Y^i \leftarrow Y_{low}$ and $snr_i \leftarrow PA\ SNR(F^i)$ with bin width Y_{low}/N; **else** $Y^i \leftarrow Y_{high}$ and $snr_i \leftarrow PA\ SNR(F^i)$ with bin width Y_{high}/N, **end if**.

11: **if** $snr_i \ge \tau_2$ & $m_i \ge t_s$, **then** $F^i \leftarrow IAN(F^i)$, with I_l iterations; **else if** $snr_i \le \tau_2$, **then** $F^i \leftarrow IAN(F^i)$, with I_h iterations, **end if**.

12: Repeat Steps 9-10 using Y^i as the initial bin width.

13: Define $\eta_i \leftarrow PA\ SNR(F^i)$ as the threshold from the first zero-bin histogram, using Y^i.

14: If first iteration, remove any pixel j with $F^i_j > 2.5 \times t_{MS}$ from data used for covariance estimate and go to Step 2. If no such pixels exist, or second iteration, go to Step 15.

15: **if** $F^i_j > \eta_i$, **then** declare pixel j anomalous, **end if**.

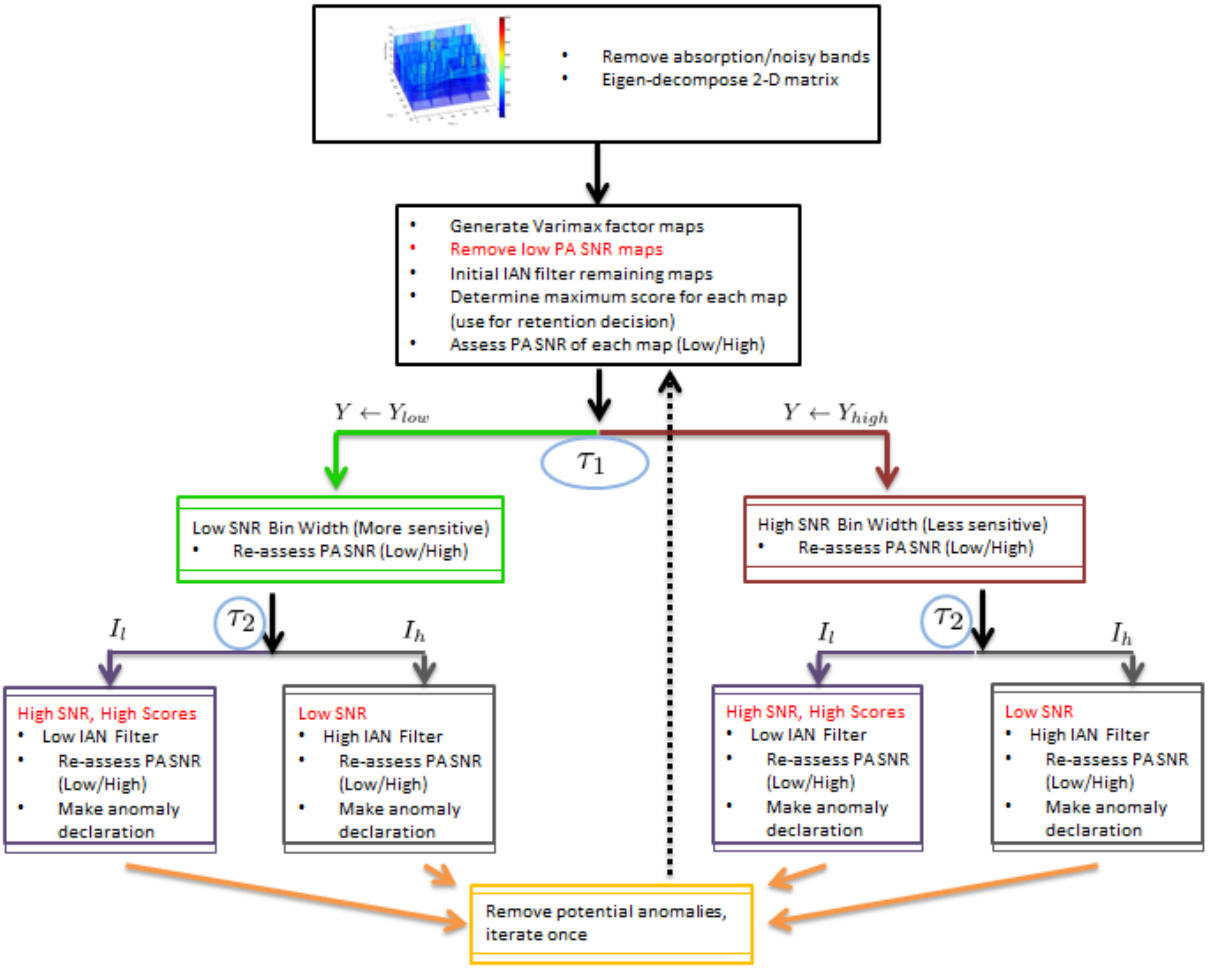


Figure 6.15: IGFAAD.

bin width is still chosen according to PA SNR in order to reduce false positives in high SNR factors and to increase true positives in low SNR factors. Step 11 is modified here, in order to take into better consideration which factors to smooth and how much to smooth them. Factors are now split into three cases at this step: 1) low SNR, 2) high SNR with low scores, and 3) high SNR with high scores. Low SNR factors are still highly smoothed to generate detection, and this is now more useful because the very low SNR factors were removed in Step 6. Thus, smoothing should mainly reveal true positives. High SNR factors with high

scores are smoothed a low number of iterations in order to reduce false positives, while not decreasing detection, as they already discriminate well. High SNR factors with low scores are not smoothed at all. This is because smoothing only serves to make anomalies more like the background in such a case. Note, the τ_1 parameters functions much like the t_{SNR} parameter used to, and now t_{SNR} is used to remove very low SNR factors. The t_s parameter is new, and is used to determine the large score criterion for high SNR factors. Again, $2.5 \times t_{MS}$ was used for the iteration screen because it was found to work well in practice.

Table 6.13 shows the settings for two experiments on IGFAAD. The first, Experiment 7, was performed to find optimal smoothing and bin parameters. Previous experiments, and additional investigation, gave that certain settings for t_{MS} and $Y_{initial}$ were near optimal. $I_{initial}$ was previously optimized to five, but investigation showed that $I_{initial} = 4$ gave similar results and provided benefit to TPF on ARES1D. The t_{SNR} threshold was chosen by looking at all of the factor mappings for the same seven training problems used previously. ARES1D was most sensitive to this threshold, but $t_{SNR} = -1$ removed the bulk of the factor mappings providing large false positives to the seven problems, without removing those mappings that could provide TPF increases. This threshold makes sense intuitively, as it implies that the potential anomaly variance is smaller than the background variance. Fixing these parameters allowed for a 3^7 full-factorial design, again fitting and optimizing Equation 6.7 as a response. The range for t_s was chosen based on investigation of the factor maps, with a number larger than 20 seeming to provide little benefit.

The optimal for Experiment 7 is also shown in Table 6.13. The reduced model had 26 significant terms at $\alpha = 0.05$ and a $R^2_{adj} = 0.9825$, with all parameters significant linearly, quadratically, or within a two-way interaction. Results at this optimal setting are shown in Table 6.14, where the training images are highlighted. Results were extremely competitive with AutoGAD and MPCA, often yielding similar or better TPFs, and lower FPFs. In fact, results were more consistent than either of the MPCA settings. The only problem where

Table 6.13: IGFAAD Experiment Settings.

Parameter	Name	Experiment 7	Opt	Experiment 8	Opt
t_{MS}	Max Score Threshold	7.05	7.05	7.05	7.05
t_{SNR}	Bin Width PA SNR Threshold	-1	-1	-1	-1
$I_{initial}$	Initial IAN Iterations	4	4	4	4
I_h	IAN Iterations High	0, 25, 50	45	0, 25, 50	40
I_l	IAN Iterations Low	0, 15, 30	12	0, 10, 20	11
$Y_{initial}$	Pixels Per Bin Initial	500	500	50	50
Y_{low}	Pixels Per Bin Low	200, 350, 500	356	10, 30, 50	10
Y_{high}	Pixels Per Bin High	500, 650, 800	540	50, 70, 90	50
τ_1	Bin Choice SNR Threshold	0, 7.5, 15	7.17	0, 7.5, 15	0
τ_2	Smoothing Choice SNR Threshold	0, 5, 10	10	0, 7.5, 15	7.48
t_s	Score Magnitude Threshold	10, 15, 20	20	20	20

MPCA showed a true advantage was the 4Ships2 image, where the FPF for IGFAAD was significantly higher. However, IGFAAD vastly outperforms MPCA on the Scene1, Ship1, and HyMAP images. In order to further improve TPFs, I_h was set to 20 and Y_{low} was set to 300. The effect of this change had been observed throughout experimentation. These changes led to the results shown in the *Exp 7 Mod* column. This greatly boosted the TPF for ARES1D while minimally affecting most of the other images. Meanwhile, this also shortened computational times, as fewer iterations of smoothing were being performed.

Experiment 8, with settings and optimal in Table 6.13, was performed to investigate the use of Equation 6.3 within the framework to choose the bin width. Recall, IGFAAD uses Y/N for the bin width, whereas that equation also adjusts according to the range of scores. Again, certain parameters were not varied because they were likely near optimal based on previous experiments. Interestingly, the smoothing iterations came out to be nearly identical to Experiment 7. The final reduced response model had 16 significant terms, $R^2_{adj} = 0.9915$, and all parameters significant in some form. Using the algorithm at these optimal settings and with the variable bin width provided the best FPFs on many problems, but at the expense of the TPF rate (thus, why these are highlighted in blue in Table 6.14).

Further, TPF rates for some of the AVIRIS images and the HyMAP image were very poor, just as in the MPCA case. Adjusting the bin width to be $Y/(2N)(max(scores) - min(scores))$ improved results slightly, but also began to noticeably increase FPFs on all of the images. This is shown as the *Exp 8 Mod* column. Upon more inspection of the AVIRIS and HyMAP imagery, the reasons for the poor performance of MPCA and the IGFAAD with a variable bin parameter (variable to the score range) became clear. These images have many more pixels than the HYDICE imagery, and also yield much larger scores. Thus, whereas the score ranges for the ARES images are typically between 20-30 at most for a factor, they can be as high as 1000 for these larger images. Therefore, it becomes

		AutoGAD	MPCA (0.1)	MPCA (0.249)	GFAAD Exp 5	IGFAAD Exp 7	IGFAAD Exp 7 Mod	IGFAAD Exp 8	IGFAAD Exp 8 Mod
ARES1F	TPF	0.9841	0.9891	0.9940	0.9662	0.9643	0.9652	0.9434	0.9811
	FPF	0.0389	0.0259	0.0259	0.0154	0.0142	0.0178	0.0102	0.0454
	Time (s)	1.1954	3.4239	3.3048	0.4996	1.2155	0.8559	1.1119	1.2314
ARES2F	TPF	0.9749	0.9902	0.9837	0.9414	0.9544	0.9642	0.9218	0.9381
	FPF	0.0567	0.0537	0.0767	0.0201	0.0245	0.0242	0.0075	0.0184
	Time (s)	2.1267	4.8439	4.1189	1.5013	4.3387	3.1814	3.1784	3.5777
ARES3F	TPF	0.8272	0.8414	0.8000	0.8414	0.8483	0.8552	0.8414	0.8483
	FPF	0.0623	0.0266	0.0194	0.0584	0.0532	0.0700	0.0184	0.0806
	Time (s)	6.2037	3.4645	3.3309	0.9667	3.8001	2.9010	2.3999	3.5728
ARES4F	TPF	0.7362	0.6789	0.6789	0.8440	0.8349	0.8349	0.7890	0.8349
	FPF	0.0527	0.0378	0.0285	0.0309	0.0282	0.0334	0.0137	0.0266
	Time (s)	4.3283	1.9803	1.7072	0.6231	2.2738	1.7703	1.8257	1.9933
ARES1D	TPF	0.9850	0.8979	0.7660	0.7489	0.8085	0.9489	0.6894	0.9957
	FPF	0.0450	0.0313	0.0257	0.0260	0.0143	0.0205	0.0118	0.0856
	Time (s)	2.3988	5.9893	5.7221	0.8016	2.6185	1.4730	1.6674	2.2882
ARES2D	TPF	0.9712	0.9522	0.7591	0.9388	0.9446	0.9446	0.8432	0.9407
	FPF	0.0327	0.0504	0.0361	0.0154	0.0143	0.0140	0.0049	0.0135
	Time (s)	7.5703	2.9470	2.7712	0.8296	3.0816	2.2803	2.3716	2.8173
ARES1C	TPF	--	--	--	--	--	--	--	--
	FPF	0.0112	0	0	0.0042	0	0	0.0020	0
	Time (s)	0.6396	2.7288	2.5232	0.3352	0.3345	0.2965	0.5510	0.3655
ARES2C	TPF	--	--	--	--	--	--	--	--
	FPF	0.0348	0	0	0.0129	0.0121	0.0125	0.0122	0.0214
	Time (s)	1.1739	2.7494	2.6025	0.3210	0.8129	0.5204	0.5582	0.7317
4Ships2	TPF	1	1	1	0.9940	1	1	0.9880	0.9970
	FPF	0.1758	0.0178	0.0173	0.1032	0.0967	0.1081	0.0272	0.0840
	Time (s)	32.5945	35.4932	33.9093	17.5601	86.0172	53.4131	58.2945	73.8429
Virgin1	TPF	0.9875	1	1	0.9875	0.9750	0.9750	0.9625	0.9750
	FPF	0.1411	0.1533	0.1180	0.0947	0.0714	0.0724	0.0441	0.0991
	Time (s)	1.1966	2.4670	2.3581	0.8268	4.3461	2.6612	2.9730	3.6322
Scene1	TPF	--	0.6000	0.5862	0.9887	0.9899	0.9899	0.7565	0.8512
	FPF	--	0.0009	0.0010	0.0375	0.0365	0.0365	0.0016	0.0026
	Time (s)	--	55.8701	54.6015	33.6119	100.7544	83.7330	63.2278	66.0144
Ship1	TPF	--	0.4029	0.3210	0.9229	0.9112	0.9112	0.2205	0.3210
	FPF	--	0.0034	0.0029	0.0668	0.0601	0.0646	0.0020	0.0044
	Time (s)	--	36.8604	35.0338	26.5833	81.3767	68.6516	53.0620	58.9394
HyMAP	TPF	--	0.4483	0.4000	0.7103	0.8759	0.8759	0.1586	0.4276
	FPF	--	0.0829	0.0814	0.0853	0.0804	0.0833	0.0292	0.0942
	Time (s)	--	14.4894	13.7981	7.6465	35.5744	22.3603	22.0789	28.2242

Table 6.14: IGFAAD Optimization Results.

much harder to provide a common, optimal bin width parameter for these images and the ARES images because they require different sensitivities relative to their score ranges. The widths that work well for ARES generate mappings for some of the AVIRIS imagery where only very extreme target or background pixels are identified. This is why Y/N works so well in the IGFAAD framework, as this sensitivity adapts appropriately according to the size of the image and is not biased by the score magnitudes. MPCA also suffers because the larger images have more significant components that get aggregated into its mappings. It should be clear that the SNR thresholding is key to IGFAAD, as the common maximum score threshold becomes less critical for those images with very high scores. In other words, one of IGFAAD's main advantages over MPCA and AutoGAD is that it better adapts to the components, and can handle scores from images that behave differently than a training set of images.

Given the analysis, and pending future, more in-depth research of the parameters and their interactions, it is recommended that the settings shown in Table 6.15 be used for an arbitrary image. These settings do have some possible trade off of TPF and FPF, but seem to work well across many image types and also provide more computational efficiency. Again, this efficiency and increased TPF can come at the cost of a slightly increased FPF. The GFAAD algorithm is extremely efficient and has fewer parameters, but the IGFAAD algorithm provides slightly better rates. In both cases, if the few bands with sensor artifacts are also removed, the FPFs improve slightly.

For a final set of results, IGFAAD with the settings from Table 6.15 was run on all of the imagery used in this research. Recall, these settings emphasize a higher TPF at possible expense of slightly increased FPF. Results are shown in Table 6.16. Here, the metrics from Section 4.4 are also included.

Only the run03m20 image was problematic, but this is not unique to the factor analysis method. MPCA with $Y_{initial} = 0.249$ achieved a TPF of 0.5954 and FPF of 0.0153 on the

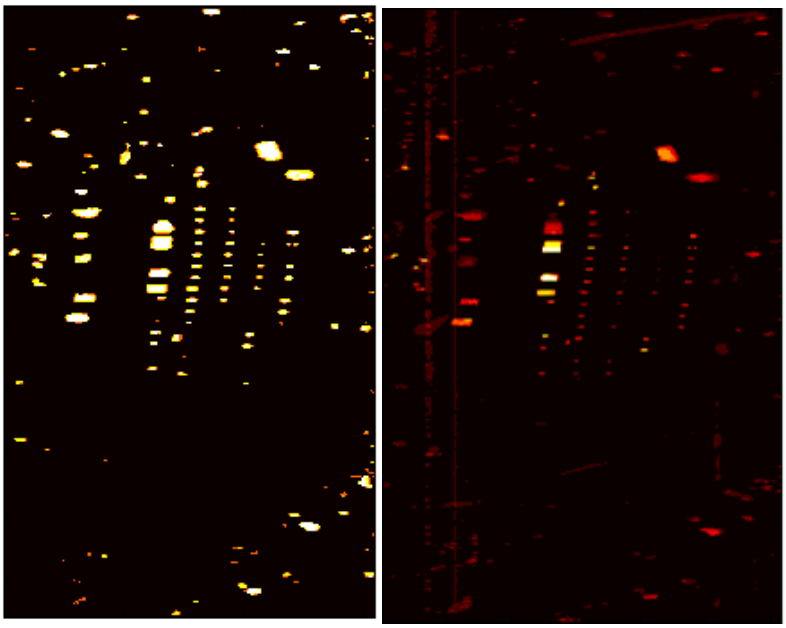
Table 6.15: GFAAD and IGFAAD Recommended Settings.

Parameter	Name	GFAAD	IGFAAD
t_{MS}	Max Score Threshold	7.05	7.05
t_{SNR}	Bin Width PA SNR Threshold	15	-1
$I_{initial}$	Initial IAN Iterations	5	4
I_h	IAN Iterations High	0	20
I_l	IAN Iterations Low	0	12
$Y_{initial}$	Pixels Per Bin Initial	500	500
Y_{low}	Pixels Per Bin Low	500	300
Y_{high}	Pixels Per Bin High	650	540
τ_1	Bin Choice SNR Threshold	(τ) 0	7.17
τ_2	Smoothing Choice SNR Threshold	N/A	10
t_s	Score Magnitude Threshold	N/A	20

	TPF	FPF	PTNB	PTIB	PBDA	Time (s)
ARES1F	0.9652	0.0178	1	1	0.0874	0.8648
ARES2F	0.9642	0.0242	0.9667	0.9667	0.1261	3.2368
ARES3F	0.8552	0.0700	0.75	0.8	0.0987	2.6053
ARES4F	0.8349	0.0334	0.8276	0.8276	0.1622	1.7478
ARES1D	0.9489	0.0205	1	1	0.0526	1.5378
ARES2D	0.9446	0.0140	0.9565	0.9565	0.0896	2.2968
ARES1C	--	0	--	--	--	0.2888
ARES2C	--	0.0125	--	--	--	0.4993
run03m20	0.4339	0.0392	0.7848	0.7848	--	34.5109
4Ships2	1	0.1081	1	1	--	54.7129
VirginIslands1	0.9750	0.0724	1	1	--	2.6085
Scene1	0.9899	0.0365	0.9565	0.9565	--	85.8087
Ship1	0.9112	0.0646	1	1	--	69.1009
Oil Spill	--	0.0612	--	--	--	59.8836
Reno	--	0.1012	--	--	--	31.1117
Red Sea	--	0.0418	--	--	--	13.8540
Pavia	--	0.0364	--	--	--	50.3347
PaviaU	--	0.0413	--	--	--	14.2296
HyMAP	0.8759	0.0833	0.8889	0.8889	0.4894	23.5060

Table 6.16: IGFAAD Imagery Results.

same image. Figure 6.16 shows these two results, where the heat map intensity is according to the number of components or factors on which the anomaly was declared. Again, the sensor artifacts are declared anomalous by IGFAAD, inflating the FPF value. Using $Y_{low} = 50$ raised the TPF for IGFAAD to a similar 0.56, but also raised the FPF further. This image performed in this manner for several reasons. First, there is a large number of targets in the image and the image is larger than the other ARES images. Additionally, these targets are more homogeneous to the background, as reflected by the improvement when lowering the bin size. Fortunately, a high percentage of the targets were still found.

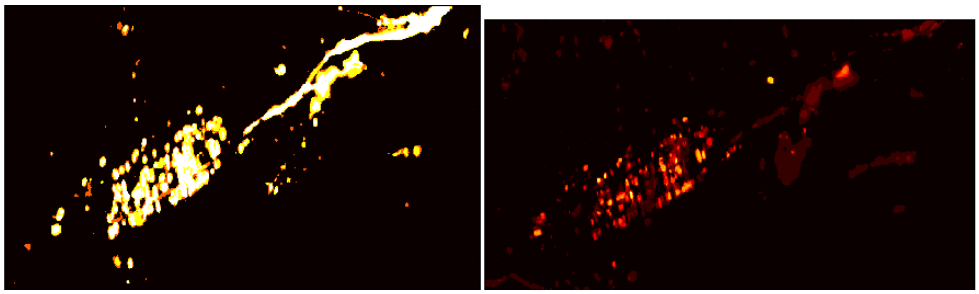


(a) MPCA. (b) IGFAAD.

Figure 6.16: run03m20 Anomaly Declarations.

The HyMAP image was a case where IGFAAD significantly outperformed MPCA, where MPCA is used as a prime comparison due to its lack of randomness. Figure 6.17 depicts the two results. We can see that although IGFAAD is more prone to finding false

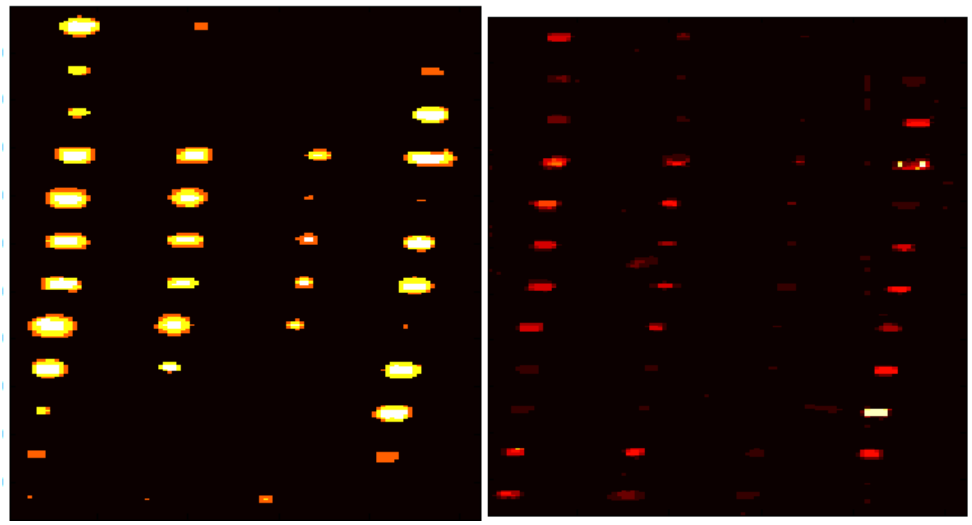
positives in background, it also focuses more precisely on the full-pixel targets. Meanwhile, MPCA identifies larger areas with anomalies contained therein, likely as a by-product of its use of aggregation of the principal components and subsequent local filtering. Similar results occurred with ARES2D, shown in Figure 6.18, and other ARES imagery.



(a) MPCA. (b) IGFAAD.

Figure 6.17: HyMAP Anomaly Declarations.

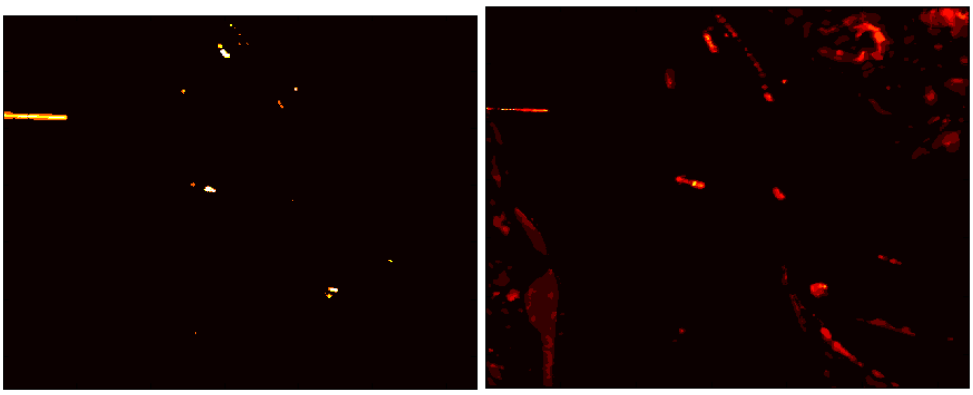
This was not necessarily the case for every image. Ship1 is shown for MPCA and IGFAAD in Figure 6.19. Here, MPCA was not sensitive enough to find a large percentage of the target pixels, while IGFAAD was perhaps too sensitive in also identifying many land pixels as anomalous due to them being highly different from the water pixels. It is clear that optimizing these algorithms to different sets of images would be ideal, but the IGFAAD algorithm appears more robust across images to a single set of parameter settings. The AVIRIS images, in general, have much higher scores and SNR values, and are larger in size. Thus, they often require differently tuned bin widths and smoothing than the ARES images to achieve best TPF and FPF rates. They also have the added complication that the land background class is significantly different in signature than water pixels. Again, to achieve absolute best TPF and FPF rates, parameter settings have to be different than the



(a) MPCA. (b) IGFAAD.

Figure 6.18: ARES2D Anomaly Declarations.

ARES imagery to account for the different image characteristics. The dynamic bin width and smoothing in GFAAD and IGFAAD help to start account for this.



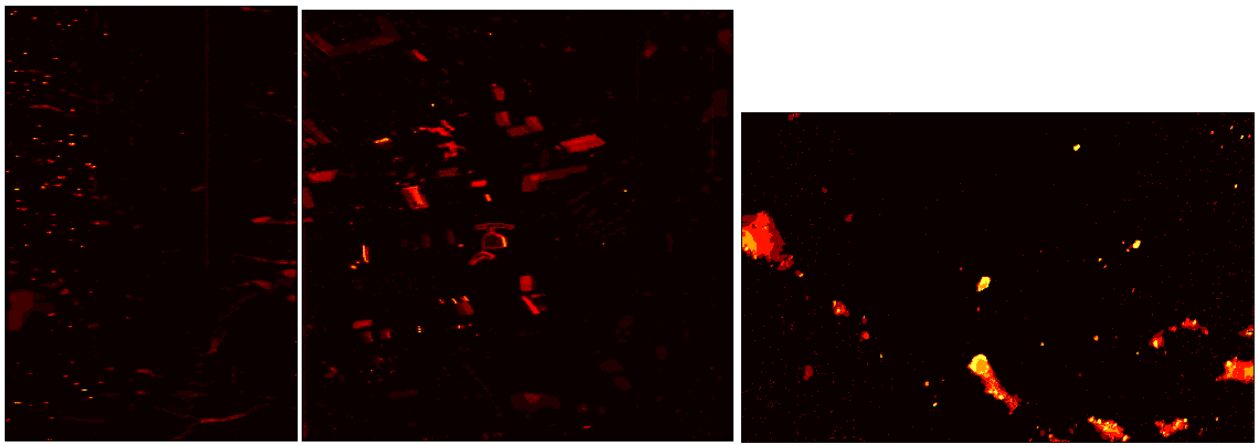
(a) MPCA. (b) IGFAAD.

Figure 6.19: Scene1 Anomaly Declarations.

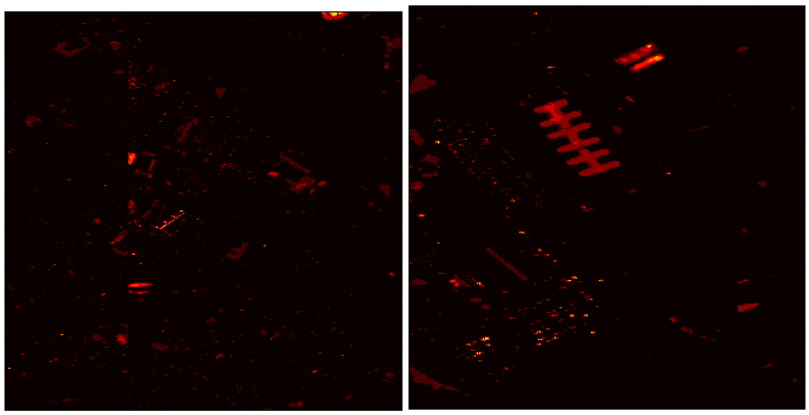
Anomaly declarations for those images without known target masks are shown in Figure 6.20 using IGFAAD. As before, intensity reflects the number of factors on which the pixel was nominated. ARES1C and ARES2C are not shown, although IGFAAD declared no targets for ARES1C. In the oil spill, what could potentially be oil slick in the water is identified. Buildings are primarily identified in the Reno and Pavia images. Some of the coral reef is identified in the Red Sea image. The full metal sheets class, as well as what appear to be cars in parking lots and a metal structure are identified in the Pavia University image.

For a final comparison, ROC curves were built for ILRX, TAD (using recommended settings from Basener and Messinger [24]), BACON, AutoGAD by varying the bin size, and MPCA by varying $Y_{initial}$. IGFAAD proved difficult to generate a relevant ROC curve for due to the significant interactions amongst parameters. Fixing $Y_{low} = Y_{high}$ and varying a single common bin width would provide a fuller curve, but this would not reflect the intended adaptive nature of the algorithm. Varying Y_{low} and I_h by themselves generated only a few points, while t_{MS} generates a large variation. The latter was varied to provide the curve shown in Figure 6.21 for ARES1F, while the former were used for ARES2D. Unfortunately, or fortunately if considering performance, due to the significance of parameter interactions these ROC curves are not necessarily representative. However, what Figure 6.21 does show is that the AutoGAD, MPCA, and IGFAAD operating points analyzed throughout this chapter are located in a very high region of the performance space relative to other algorithms.

GFAAD and IGFAAD show great promise, and are very competitive with existing algorithms. They appear to be more robust to different image types than existing algorithms, while maintaining a high TPF and relatively low FPF. Additionally, the mappings they generate and use to identify anomalies have a corresponding meaning to the materials in the image, are deterministic, and are relatively interpretable as a result of using



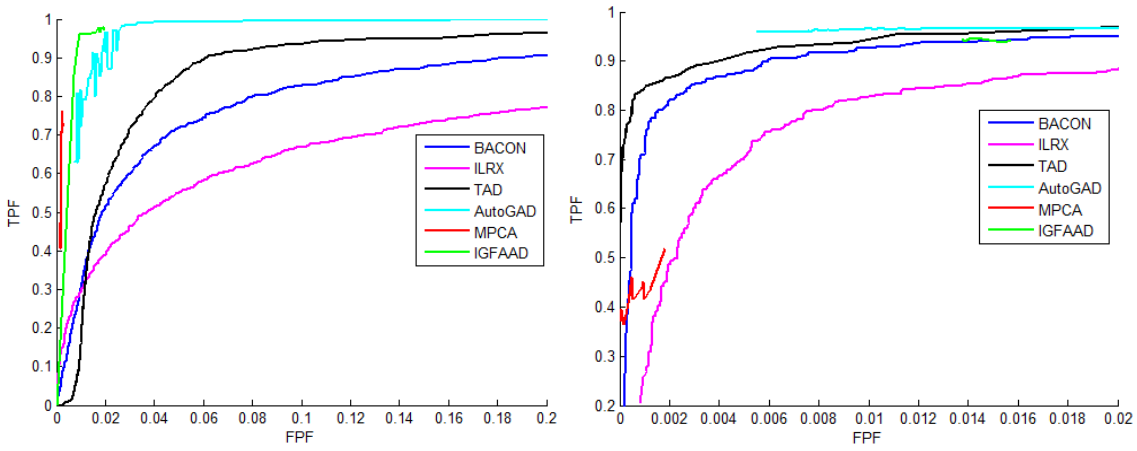
(a) Oil Spill. (b) Reno. (c) Red Sea.



(d) Pavia. (e) Pavia Univ.

Figure 6.20: IGFAAD Anomaly Declarations.

factor analysis and having highly loaded bands. However, to fully reach potential, further investigation into the interactions of the many aspects of the algorithms is warranted. A final SNR filter at the end of each algorithm, to remove factors that yield false positives after smoothing, may also be warranted. Unfortunately, again, this is not straightforward due to significant parameter interaction and due to the differing SNR characteristics of factors generated from different images. Next, a non-linear form of this factor analysis-



(a) ARES1F. (b) ARES2D.

Figure 6.21: ROC Comparisons.

based framework is developed in order to utilize the better discrimination that a non-linear mapping can provide.

VII. Large-Scale Kernel Principal Component Analysis

7.1 Literature Review

7.1.1 Eigen-Decomposition.

Kernel Principal Component Analysis (KPCA) can be problematic, in that the Gram matrix is $N \times N$ in size for a dataset with N exemplars. Thus, for example, in order to build global KPCs for an image with N pixels, eigen-decomposition of the $N \times N$ similarity matrix has to occur. Obviously, for even moderately sized images or data sets this is prohibitive, despite the fact that only a subset of the $r < N$ non-zero eigenvalues and corresponding eigenvectors may be of interest.

The power method and Lanczos method are both iterative methods that approximate eigenvectors of a square matrix. Each typically require some form of matrix vector multiplication that with large N can be computationally expensive. Lanczos can be subject to roundoff error, but can be very useful if a matrix is sparse [79]. For Gram matrices, this is typically not the case, although they do have the nice properties of being symmetric and normal. Standard methods to find the eigenvectors take $O(N^3)$ time, which is prohibitive for large N. Schölkopf, Achlioptas, and McSherry [184] proposed random sparsification and random rounding methods to speed computation of the kernels and KPCs, but these can be subject to higher error with large data sets.

The Nyström method approximates the Gram matrix K by sampling $m << N$ columns from K [139]. This generates a sample of columns from K, or a sub-matrix A that can rearranged such that K and A are written as

$$A = \begin{bmatrix} W \\ S \end{bmatrix} \text{ and } K = \begin{bmatrix} W & S^T \\ S & B \end{bmatrix}, \tag{7.1}$$

where W is $m \times m$, S is $(N - m) \times m$, and B is $(N - m) \times (N - m)$. If the singular value decomposition (SVD) of W is $U\Sigma U^T$, where the singular values σ_i of W are in non-

increasing order on the diagonal of Σ, then the rank $r \leq m$ Nyström approximation for K is,

$$\tilde{K}_r = A W_r^+ A^T. \tag{7.2}$$

Here, $W_r^+ = \sum_{i=1}^{r} \sigma_i^{-1} U^{(i)} U^{(i)T}$ and $U^{(i)}$ is the i-th column of U. W_r^+ is also referred to as a Moore-Penrose pseudo-inverse [67]. This approximation is $O(Nmr + m^3)$ but also requires large m to ensure sufficient sampling [139]. Drineas and Mahoney [67] developed a data-dependent non-uniform sampling methodology for the columns that ensured the bound

$$\|K - A W_r^+ A^T\|_\zeta \leq \|K - K_r\|_\zeta + \epsilon \sum_{i=1}^{N} K_{ii}^2, \tag{7.3}$$

where the norm could be L_2 or Froebenius, with high probability if choosing $O(r/\epsilon^4)$ columns. For small ϵ and large eigenvalues this bound becomes less useful. The sampling of columns is not computationally prohibitive, as at no point does the full Gram matrix have to be generated.

Halko, Matinsson, and Tropp [92] developed a randomized SVD algorithm to estimate the kernel eigenvectors that they showed performed well in practice, but that requires a complete pass over the data. Specifically, given scalars r, q_1, and q_2, let $Y = K^{q_2-1} K Z$, where Z is a $N \times (r + q_1)$ standard Gaussian random matrix. An orthonormal matrix Q is found by QR decomposition such that $Y = QQ^T Y$. Next, SVD is performed on $Q^T W Q = V \Lambda V^T$ to yield the eigenvector estimates QV and eigenvalue matrix estimate Λ for K.

Li, Kwok, and Lu [139] combined the randomized SVD and Nyström methods in order to retain the efficiency of Nyström while also retaining the accuracy of the randomized SVD. In their algorithm, they sampled m columns of K uniformly at random without replacement, and built W as in Nyström. Next, they used random SVD on W to yield a set of eigenvectors V and an eigenvalue matrix Λ, using $q_1 = 5$ and $q_2 = 2$ in practice.

These could be used to reconstruct an approximate $\tilde{K}$ as

$$\tilde{K} = \left(\sqrt{\frac{m}{N}} AV\Lambda^{+}\right)\left(\frac{N}{m}\Lambda\right)\left(\sqrt{\frac{m}{N}} AV\Lambda^{+}\right)^{T}. \tag{7.4}$$

For this reconstructed Gram matrix $\tilde{K}$, they showed that,

$$\mathbb{E}\|K - \tilde{K}\|_2 \leq \zeta^{1/q_2}\|K - K_r\|_2 + (1 + \zeta^{1/q})\frac{N}{\sqrt{m}} \max_i K_{ii} \tag{7.5}$$

where $\zeta = \left(1 + \sqrt{\frac{r}{q_1 - 1}} + \frac{e\sqrt{r + q_1}}{q_1}\sqrt{m - r}\right)$. This method also saved time complexity, in only requiring $O(Nmr + r^3)$ operations. They showed favorable approximation error and computational time on data sets including MNIST versus the standalone Nyström and randomized SVD algorithms, and an ensemble method [139]. Although their algorithm was unnamed, for this research it is referred to as *NyApprox*, denoting Nyström Approximation. The rank-r approximation can also be used to generate approximate scores. In the case of *NyApprox*, these are computed as $AV_l\Lambda_l^{-1/2}$ for the leading $l \leq r$ eigenvectors, where the $N - 1$ constant from the covariance is often ignored because it only scales values [70, 168, 215].

Thus far, these eigen-decompositions and scores are relative to the original Gram matrix, and not its centered version, which is required to accurately model the covariance in the higher-dimensional space for purposes of KPCA. Decomposing the original matrix, *i.e.*, $K = \Phi(X)\Phi(X)^T$ is also referred to as solving the dual problem, while decomposing $\Phi(X)^T\Phi(X)$ is the primal [168]. In order to easily convert the low-rank solutions from the dual to the primal, consider the $N \times r$ scores matrix, $E = AV_r\Lambda_r^{-1/2}$. K can be equivalently approximated as $\tilde{K} = EE^T$. The full Gram matrix is centered by,

$$\hat{K} = HKH, \tag{7.6}$$

where $H = I - \frac{1}{N}\mathbf{1}_N^N$, I is the $N \times N$ identity matrix and $\mathbf{1}_N^N$ is a $N \times N$ matrix of ones [228]. Given the rank-r approximation EE^T to K, $\hat{K}$ can be approximated using $HE\,(HE)^T$.

Therefore, the leading l score vectors can be centered as,

$$\hat{E} = HAV_l\Lambda_l^{-1/2}. \tag{7.7}$$

Equivalently, the Nyström approximation can be performed on the appropriately centered versions of Equation 7.1, where A and W are centered by subtracting the kernel column means. Note, this entire process is very similar to the development previously shown in Section 3.3, where here $\hat{W}$ is similar to the centered training kernel matrix and $\hat{S}$ is similar to the centered test kernel matrix, but where they were previously centered only in regard to the training matrix. In other words, with the Nyström method the goal is to try and best approximate the full N-exemplar eigenvector, while using a subset of exemplars as a training matrix only uses the corresponding m-exemplar eigenvector as an approximation. Nonetheless, these are very similar as they both use a m-exemplar skeleton to approximate the desired kernel eigenvectors.

Other Nyström approximations also exist, such as greedy sampling where columns are sampled based on constructing the best rank-1 approximation for a current residual matrix [70]. However, the *NyApprox* and clustering based approximations discussed in Section 7.1.2 are used primarily due to their efficiency and their published, generally better performance [70, 139, 228].

Other approximation methods also exist for the kernel eigenvectors. Kim et al. [124] developed an iterative KPCA method, or Hebbian algorithm, using a learning rate for reconstruction. Gunter et al. [85] tried to enhance this by adding a gain vector, but their algorithms still required hours to converge on data sets such as MNIST. As MNIST is not necessarily larger in dimension than HSI data, it is clear that these approaches are not desirable here.

7.1.2 Landmark Points.

Landmark points, the building of which are referred to synonymously in this research as *skeleton generation*, are exemplars taken or derived from the data and are meant to

represent the data in some way. Referring back to the Nyström approximation and its use of a subset of columns, the exemplars corresponding to the selected columns are generally equivalent to landmark points, except that they are chosen randomly whereas landmark points are typically chosen according to criteria.

Zhang and Kwok [228] developed the Nyström approximation using k-means centroids to provide the W sub-matrix, using the centroids as landmark points. Similarly, Kwon and Nasrabadi [133] used k-means centroids of the data to represent background, and to provide a kernel version of the RX algorithm using the covariance KPCs corresponding to the centroids for a kernel Mahalanobis distance. This was previously discussed in more detail in Section 3.11.1.3.

Brandes and Pich [38] used landmark points in Multi-Dimensional Scaling, and found that maximizing the minimum distance to previous landmarks as more landmarks were added worked well in constructing their projections. Chen and Cai [52] used both cluster centroids and random exemplar selection for landmark points in spectral clustering.

As with Kwon and Nasrabadi's [133] centroid approach to kernel Mahalanobis distance, landmark points generated in any manner can be used explicitly to build the KPCs. The kernel scores then reflect projections onto the directions of variance for the landmark points in the higher-dimensional feature space. With a good choice of landmark points, such as those well representing the background, these projections should reveal anomalous pixels.

Landmark points can also be integral to Affinity Propagation (AP) for large data. Recall from Section 3.9.3, that several $N \times N$ matrices are used to generate the representative exemplars. This can clearly be computationally prohibitive. To aid in finding centers in very large data sets, Xia et al. [220] used landmark points to develop a global Landmark Affinity Propagation (LAP) algorithm. Specifically, they randomly sampled $m < N$ exemplars from the dataset and applied AP to them, where the centers found

were a subset of the m landmark exemplars. Xia et al. [220] also developed a Partition Affinity Propagation (PAP) algorithm designed to utilize local information in order to speed efficiency. Here, they split the input similarity matrix into m subsets of exemplars and performed AP on each, resulting in m local availability matrices. These were then used to yield an initial sparse global availability matrix, where exemplars from different subsets had zero availability. This initial, sparse availability matrix was then used in AP on the entire dataset. Both of these methods inspired the two large-scale AP approaches developed shortly in Section 7.3.

7.1.3 Optimal Kernels.

KPCA, and any kernel method for that matter, is subject to the choice of kernel. Different kernels can yield very different projections and decision boundaries. In the application of HSI anomaly detection, often when KPCA is used the kernel is chosen experimentally. For example, Chunhui, Yulei, and Feng [58] varied the spread parameter σ in a Gaussian kernel in order to better detect anomalies in a KICA RX-based detector. Kwon and Nasrabadi [133] did the same for KRX.

Given a Gaussian kernel and class information, Wang et al. [211] used the KFDA criterion (previously discussed in Section 3.6) to provide a way to optimize the σ parameter. They optimized the function,

$$\sigma_{opt} = \arg\max_{\sigma} tr\left(S_W^{-1} S_B\right), \tag{7.8}$$

in order to best separate the class means with minimal within-class variance in the kernel space, where the scatter matrices S_W and S_B were as defined in Equation 3.26. They also derived a partial derivative for J with respect to σ so that the problem could be solved with a quasi-Newton algorithm.

Kim, Magnani, and Boyd [125] developed a slightly more general optimal kernel algorithm for two classes, but still using KFDA. Rather than optimizing a single kernel, they proposed optimizing over a convex set of kernels. Zhu et al. [231] used a similar idea

in kernel Canonical Correlation Analysis, using a polynomial kernel for global estimation and a Gaussian for local. In Kim, Magnani, and Boyd's method [125], they noted that the optimal kernel projection vector using the kernel Fisher ratio is known, and so the maximum achievable kernel discriminant ratio for a given kernel and training set can be explicitly calculated. This is,

$$\frac{1}{\gamma}\left[a^T KJ\left(\gamma I + JKJ\right)^{-1} JKa - a^T Ka\right], \tag{7.9}$$

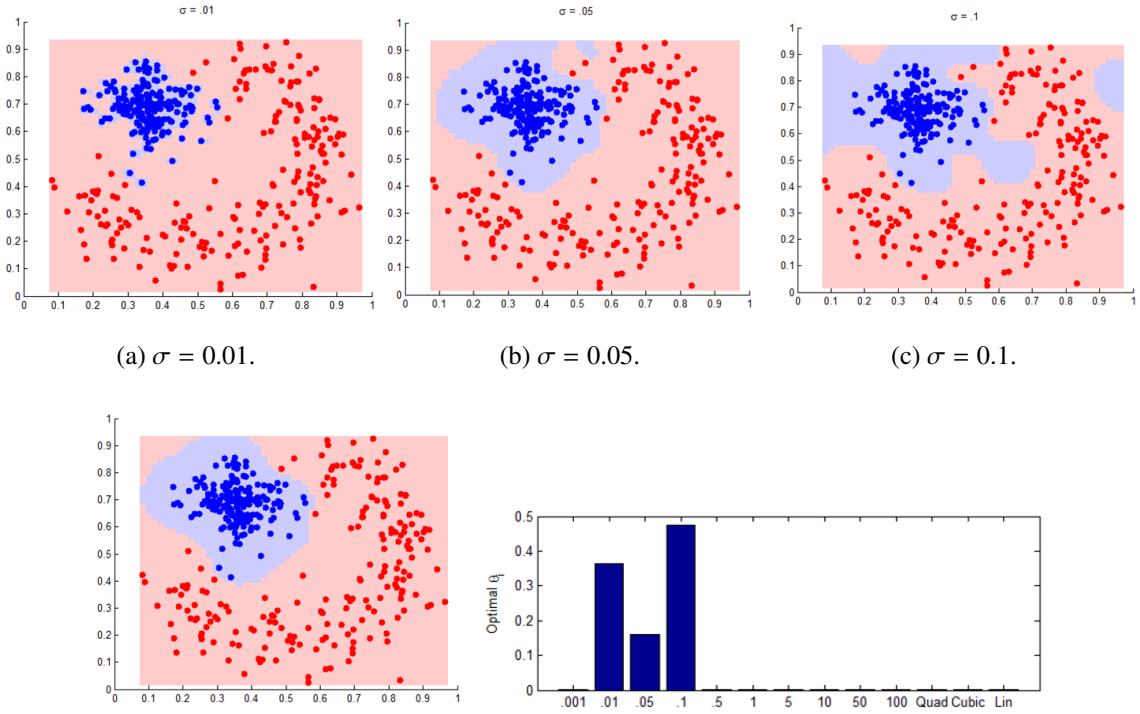
where γ is a very small regularization parameter, I is the identity matrix, K is the $m \times m$ kernel matrix for the m training exemplars,

$$a = \begin{bmatrix} (1/N_1)\mathbf{1}_{N_1} \\ -(1/N_2)\mathbf{1}_{N_2} \end{bmatrix}, \tag{7.10}$$

N_1 and N_2 are the number of exemplars in each class, $\mathbf{1}_{N_1}$ is a $N_1 \times 1$ vector of ones, and

$$J = \begin{bmatrix} \frac{1}{\sqrt{N_1}}\left(I - \frac{1}{N_1}\mathbf{1}_{N_1}\mathbf{1}_{N_1}^T\right) & 0 \\ 0 & \frac{1}{\sqrt{N_2}}\left(I - \frac{1}{N_2}\mathbf{1}_{N_2}\mathbf{1}_{N_2}^T\right) \end{bmatrix}. \tag{7.11}$$

Therefore, for $\tilde{K} = \sum_{i=1}^{c} \theta_i K_i$ with $\sum_{i=1}^{c} \theta_i = 1$, *i.e.*, a convex set of c candidate kernels, an optimal kernel can be found by plugging $\tilde{K}$ into Equation 7.9 and minimizing (it is formulated as a minimum) over the weights $\boldsymbol{\theta}$ while enforcing non-negativity and the convexity constraint [125]. This gives the optimal kernel $k^*(x, y) = \sum_{i=1}^{c} \theta_i^* k_i(x, y)$. There are two limitations to this method. First, class information is required. Second, formulation of the Gram matrices becomes expensive as the number of exemplars under consideration grows. An example of using this optimization technique is given in Figure 7.1 for the Banana Dataset and a set of thirteen kernels. Here the dataset is small enough that all data could be used in the training set, but for larger data sets the support vectors greatly define the boundaries found. This makes a good estimate for a single class necessary to ensure the support is not too small or too large. Three of the candidates are shown to depict the effect of various Gaussian kernels.



(a) $\sigma = 0.01$. (b) $\sigma = 0.05$. (c) $\sigma = 0.1$.

(d) Optimal σ. (e) Optimal θ_i.

Figure 7.1: Optimal Kernel Example.

7.1.4 Further Algorithmic Considerations.

When using kernel methods, several issues can affect results. As mentioned, the choice of kernel can drastically affect results. In Section 3.3 a few of the more common kernels were presented, to include the Gaussian, linear, polynomial, inverse multi-quadric, and hyperbolic tangent kernels. Liang and Lee [142] showed that higher-order polynomial kernels are very sensitive to outliers via general eigen-analysis.

Note that with the exception of the linear kernel, all have associated parameters themselves. In general, any constant parameters are set to zero in this research. The spread parameter σ is of particular interest. Not to be confused with singular values, this parameter defines the decay rate for several kernel types. As such, it needs to be

smaller than the size of the decision space. Even when optimizing the kernel as in Section 7.1.3, candidates are necessary. Nielsen and Canty [168] suggested that σ be larger than a typical distance between exemplars. Shi and Malik [190] used $10-20\%$ of the range of the distances between exemplars. Kwon and Gurram [132] expanded the concept of an optimal set of bandwidths for KRX by first using cross-validation on probable background pixels and simulated anomalies to find a single σ^*. The simulated anomalies were generated by adding noise to probable background pixels, and σ was optimized relative to increasing the probability of detection and minimizing false alarms. Then, an individual σ_i for each band within the kernel calculation was determined by scaling σ^* according to the variance of the band. Regardless of how kernel parameters are chosen, correct values can also depend on the technique they are being applied within and what the scaling of the data is.

Yet another consideration is how to handle singular or near singular K or W. In such a case, it is common to replace K (or similarly W) by $K + cI$ where I is the identity matrix and c is a very small constant [215]. As just alluded to, the scale of the data can also play a role in a KPCA algorithm, as the dot product of vectors with large numbers can result in extremely high similarity values that may cause issues in eigen-decomposition. To aid in mitigating this for HSI, Kwon and Nasrabadi [133] scaled the data by the largest overall radiance value found in the image.

7.1.5 Choosing Discriminating Components.

Even once the KPCs are found or approximated, if the desire to use them as mappings on which to identify anomalies, there needs to be a way to identify useful KPCs. Some techniques were discussed in Chapter 6, but warrant discussion again here, as the nature of KPC can sometimes make it more difficult to find meaningful components. In the supervised case, the best component/eigenvector could potentially be chosen using the LDA-equivalent criterion, $\frac{tr(S_B)}{tr(S_W)}$, based on the scores or its kernel counterpart [212]. However, this is not useful in an unsupervised setting.

Chiang, Chang, and Ginsberg [55] used projection pursuit to find optimal projections of HSI for anomaly detection. In projection pursuit, data is projected into a lower-dimensional space while retaining some information of interest via a Projection Index (PI). In particular, κ_3^2, κ_4^2, $\kappa_3^2+\kappa_4^2/12$, and $(\kappa_3\kappa_4)^2$ were used for their PIs, where κ_3 is skewness and κ_4 is kurtosis. After whitening the data, they used an evolutionary algorithm to optimize the projection for a given PI. They then used a first zero-bin detection histogram to threshold the projection and find anomalies. Gu, Liu, and Zhang [84, 145] also utilized skewness and kurtosis in an effort to find effective components for anomaly detection. With RX in mind, they noted that if the data in local windows followed a Gaussian distribution then the skewness and kurtosis would be zero. They also noted that if there were anomalies in the local windows, then the data would not be Gaussian distributed and the absolute values of the skewness and kurtosis would be large. They defined thresholds to devise a Local Singularity (LS) rule. Specifically, let $\theta_s >> 1$ and $\theta_k >> 1$ be constants, $\tau_s = \sqrt{6/n}$, and $\tau_k = \sqrt{24/n}$ where $n = n_1 \times n_2$ is the size of the local window. Then the respective thresholds for absolute skewness and kurtosis are defined as $T_s = \tau_s\theta_s$ and $T_k = \tau_k\theta_k$ [145].

These metrics make sense for the RX algorithm, where a Gaussian assumption is used, but as HSI data is generally not Gaussian, it is not useful for a general detector. Further, in this research, global detectors are of high interest in order to maintain efficiency given that kernel methods are more computationally expensive than their linear counterparts. A projection with a low global LS can be eliminated as there are no potential anomalies, but a projection with high global LS does not insinuate a good mapping for anomaly detection. The specific thresholds also have to be chosen.

Chen and Qian [50] used entropy values to measure information content on image channels. They defined image entropy as,

$$-\sum_{i=1}^{n} p_i log_2 p_i, \tag{7.12}$$

where n was the number of grey levels and p_i was the probability that the grey level i occurred. This metric, like LS, does not necessarily yield good projections because noisy projections can yield a discriminating value for such a metric. Johnson's PA SNR metric [110] in combination with a zero-bin histogram, discussed heavily in Chapter 6, is another potential way to identify useful KPCs by finding those where a subset of pixels are significantly different from the rest.

Izenman and Shen [106] proposed using minor kernel PCs to detect outliers, as outliers are a common source of noise. To devise a fixed threshold, they only used the Gaussian kernel. This enabled them to use $\lambda = 1$ as a cut-off for 'large' vice 'small' components, where λ_k and λ_{k+1} were the eigenvalues surrounding 1. Next, the subset of eigenvalues $\lambda_{k+1}, \ldots, \lambda_N$ were taken as candidates. As many of these may be zero, only those values and corresponding KPCs that explained $\geq 0.01\%$ of the variance were retained, where the set of all considered variance proportions was $S = \left\{ \frac{\lambda_{k+1}}{\sum_{i=1}^{N} \lambda_i}, \ldots, \frac{\lambda_N}{\sum_{i=1}^{N} \lambda_i} \right\}$. This gave a subset $S' = \{\lambda_{k+1}, \ldots, \lambda_{N'}\}$, where N' was the index of the smallest eigenvalue explaining at least 0.01% of the variance. Next, a threshold was constructed as,

$$t = \frac{\left(\sum_{i=k+1}^{N'} \lambda_i \right) / (N' - k)}{\sum_{i=1}^{N} \lambda_i}. \tag{7.13}$$

This approximated the average proportion of variance explained by the small KPCs. The smallest selected KPC on which to find outliers then corresponded to the first element of S' above t. The second smallest selected corresponded to the second element of S' above t, and so forth [106].

Although this cut-off appears promising, it does not extend well to KPCA on HSI for several reasons. First, both whether the image has been scaled and the particular σ influence if an eigenvalue as high as $\lambda = 1$ exists. Second, even with the correct scaling there may be very few eigenvalues greater than one. $k = 200$ with k-means for a landmark

set on ARES1F yielded only two eigenvalues greater than one. Therefore, nearly every eigenvector was minor. Finally, even if using a more dynamic cut-off such as MDSL from which to start, the minor KPCs for HSI data often are highly noisy. This observation is made here based on a great deal of experimentation on different images and skeleton types. Even medians or squared sums of the minor components provided little more than noise. In those cases where some discrimination was present, it often also appeared in one or more of the major components.

7.2 Approximate Kernel Factor Analysis

In order to also formulate a kernel factor analysis (KFA) method, recall the development in Section 3.4. The loadings in the linear case with a PC solution were $\hat{L} = \Lambda^{1/2}T$, where T were the eigenvectors of the covariance C and Λ was a diagonal matrix of the corresponding eigenvalues. The loadings in the non-linear case using a KPC solution are then

$$\hat{L} = \Lambda^{1/2}T = \Lambda^{1/2}\Lambda^{-1/2}\hat{\Phi}(X)^T D = \hat{\Phi}(X)^T D, \tag{7.14}$$

where D are the eigenvectors of $\hat{K}$ and T are the eigenvectors of the primal covariance problem. D can similarly be rotated using a Varimax rotation, where now the high loadings reflect emphasis on specific exemplars or landmark points [86]. Let this rotated matrix be denoted as $\hat{D}$.

In the linear case, the unweighted least squares solution for factor scores was $\left(\hat{L}^T\hat{L}\right)^{-1}\hat{L}^T\hat{X}^T$. In the non-linear space this yields,

$$\left(\hat{D}^T\hat{\Phi}(X)\hat{\Phi}(X)^T\hat{D}\right)^{-1}\hat{D}^T\hat{\Phi}(X)\hat{\Phi}(X)^T = \left(\hat{D}^T\hat{K}\hat{D}\right)^{-1}\hat{D}^T\hat{K}. \tag{7.15}$$

This can be problematic for a few reasons. First, whether using Nyström or a skeleton to approximate $\hat{K}$, it is not desirable to construct the actual $\hat{K}$ or its approximation in full if there is a large number of exemplars. This reconstruction is limited computationally by the matrix-matrix multiplication. Furthermore, estimates for D are based on either

a training set or low-rank approximation. Therefore, instead, the factor scores can be approximated using the eigenvectors V of $\hat{W}$, where this matrix is the kernel matrix for the m landmark points or chosen columns from the initial approximation. This is not as exact in constructing the factors or scores, but should still provide useful factor mappings if the skeleton is sufficient. Thus, V is $m \times l$ if l factors were retained, and $\hat{W}$ replaces $\hat{K}$. However, scores for every exemplar, and not just the training data, are also necessary in the application for this research. To do this, the latter $\hat{K}$ can be replaced by the $m \times N$ kernel matrix A^T. In other words, again rotating V to provide $\hat{V}$, this yields

$$\left(\hat{V}^T \hat{W} \hat{V}\right)^{-1} \hat{V}^T \hat{A}^T. \tag{7.16}$$

This yields a $l \times N$ matrix of scores, where centering is also taken into account. Again, as the loadings now emphasize specific exemplars, the success becomes very much dependent on the choice of skeleton with which V and $\hat{W}$ are generated.

7.3 Skeleton Generation

For results in this chapter, runs were on an Intel^{TM} Core i7 CPU Q840@1.87 GHz, 64-bit OS, with 8 GB RAM unless otherwise denoted. Again, in this research, *skeleton generation* is defined so as to be synonymous with landmark point generation. The Nyström approximation can use these skeletons to build low-rank approximations to the full Gram matrix and eigen-decomposition for a dataset, while the skeleton can more directly be used as a training set to generate KPCs against which exemplars not in the skeleton (test set) can be projected. These two methods are highly related, but the Nyström technique tries to approximate the full N-exemplar kernel eigenvectors. In order for a skeleton to be useful, it needs to represent characteristics of the full dataset in some manner. Typically this means encapsulating the principal directions of variance of the full dataset or, in the anomaly detection problem, background directions of the data against which anomalies are obvious. In this research, clustering is considered as a primary means of skeleton generation. Large-

scale clustering approaches are presented next, followed by analysis of the skeletons they yield on various data sets and the resulting KPC mappings.

7.3.1 Development of Large-Scale Skeleton Approaches.

One large-scale k-means clustering approach was already developed in Section 3.9.1, for input to the large-scale X-means and BIC-means methods discussed in Section 3.9.2. There, computational speed-ups were added to the k-means framework for use as a stand-alone algorithm and in X-means, to include using PC scores if the number of clusters k is less than the number of features p. Robust forms to make cluster assignment more deterministic and accurate were also developed by incorporating the refinement strategy of Bradley and Fayyad [37].

X-means and BIC-means are advantageous in that k is found by optimizing a BIC criterion. However, the k found can be much larger than desired for computational or operational purposes. With HSI data, this is very relevant because larger k take substantially more time to evaluate and compute. Li, Prasad, and Fowler [141] navigated the BIC issue by using a maximum number of clusters, ten, for which to estimate the BIC criterion, and calculating a BIC difference between different cluster numbers,

$$BIC_{diff} = \frac{\|BIC(i) - BIC(i-1)\|_1}{\|BIC(i)\|_1} \leq \tau_{BIC}, \tag{7.17}$$

where τ_{BIC} was a threshold set to 0.02 for the Pavia University data. Their optimal number of clusters was then the smallest i that minimized BIC_{diff} under $\tau_{BIC} = 0.02$. In the case of the Pavia University image, this was seven. This approach is still problematic in that the threshold is subjective and a range of possible k has to be chosen. Here, X-means and BIC-means are used, initially, with an upper bound on k that serves only to limit the computational expense.

Affinity Propagation for large-scale data also warrants special attention for purposes of generating a skeleton. The algorithm is of interest as a competitor to the k-means approaches because resulting centers are actual data exemplars and k is still chosen

automatically. Unfortunately, the algorithm requires storage and manipulation of three $N \times N$ matrices. A first approach to mitigating this is to use Xia et al.'s LAP algorithm [220]. However, instead of randomly choosing a skeleton, a version of Brandes and Pich's [38] landmark approach is used. That is, to generate the skeleton that is input to AP (the set of candidates for centers), first the exemplar nearest the median of the data is identified as the first landmark. Next, until a desired number of landmarks are found, the exemplar with the furthest minimum distance to any of the current landmarks is added to the skeleton. L_2 is used for distance, given the prior analysis in Chapter 4. This skeleton generation and subsequent use of Xia et al.'s LAP algorithm [220], is referred to as LAP in this research.

A second approach, using the idea of partitioning but distinct from the PAP algorithm, is shown as Algorithm 7.1. Here, a number of unique subsets of the data are chosen based on a desired subset size. This size parameter is primarily to help with computational burden, but also gives an upper bound on how many centers the algorithm yields. These subsets are not exactly a partition, as a small number of exemplars are not considered at all, unless N is exactly divisible by m. This is done partially for implementation purposes, but as HSI images and other very large data sets are such that any non-outlier exemplar should have a similar exemplar elsewhere in the data, removing a small number of exemplars from consideration does not affect the end result greatly. Once the subsets are chosen, according to uniform random sampling, each subset is clustered using AP. The resulting centers across all subsets are combined into a single set of candidate centers. Now, if the number of candidates is too large, then the partitioning and AP process is repeated on the candidates until either there is no change in candidates or until $\leq m$ candidates remain. Finally, AP is performed on the candidates to yield a final set of centers for the data. Essentially, this Partition Landmark Affinity Propagation (PLAP) algorithm forms a minimal skeleton estimate for the large-scale dataset by forming skeletons for subsets of the dataset, and then forming a skeleton of the skeletons. This is beneficial because it allows some control

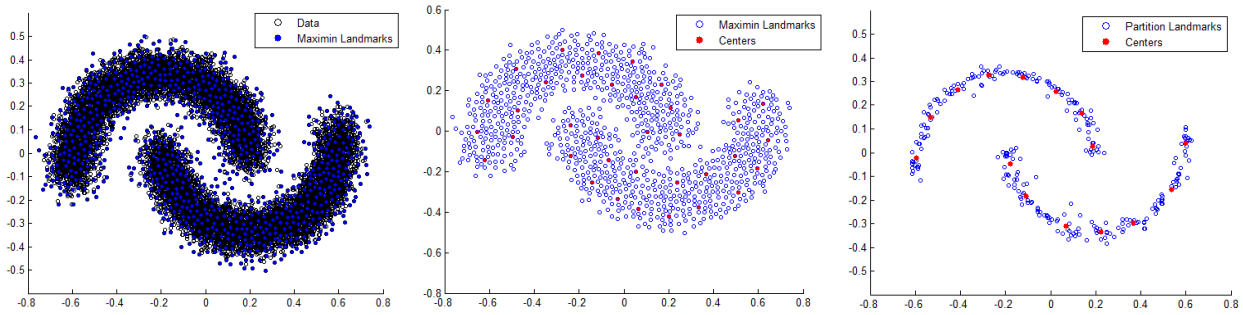
over the number of centers in the skeleton, and it fuses information across the subsets used initially. Examples of the stages of landmarks for the LAP and PLAP algorithm are shown in Figure 7.2 for the Half-Moon dataset, where a Gaussian kernel was used for similarity.

Algorithm 7.1 Partition Landmark Affinity Propagation (PLAP)

1: Let m be a desired number of exemplars for each initial subset, and overall.
2: Randomly sample without replacement to provide $\lfloor N/m \rfloor$ subsets of the original $N \times p$ dataset X.
3: **for** $i = 1 : m$ **do**
4: $C_i \leftarrow AP(X_i)$, where X_i is the i-th subset and C_i is the corresponding set of estimated centers. This performs affinity propagation on each subset.
5: **end for**
6: $C \leftarrow \cup_{i=1}^{m} C_i$.
7: **if** $|C| > m$, where $|C|$ is the number of centers **then**
8: Do Steps 1-6, using $N \leftarrow |C|$. This yields the centers C'. $C \leftarrow C'$.
9: **if** $|C| > m$ and at least one center in C changed **then**
10: Go to Step 8.
11: **else** Proceed.
12: **end if**
13: **else** Proceed.
14: **end if**
15: $C \leftarrow AP(C)$.

7.3.2 Skeleton Analysis.

In order to evaluate how to choose the best skeleton with which to generate the KPCs for HSI, a starting point is to evaluate methods on lower dimensional data sets from Chapter



(a) LAP: Initial Landmarks. (b) LAP: Final Landmarks. (c) PLAP: Initial & Final Landmarks.

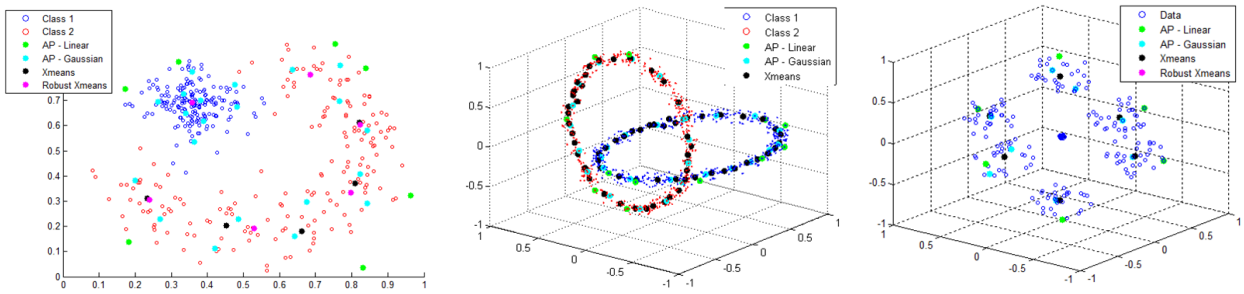
Figure 7.2: LAP and PLAP Half-Moon Example.

2. Figure 7.3 shows the data and centers for three of the lower-dimensional data sets, where a linear and Gaussian kernel were used to build similarities for AP. For the Gaussian, σ was set as a function of the average distance of exemplars from the mean of the data. Specifically,

$$\sigma = \sqrt{\left(\sum_{i=1}^{N} \|x_i - \bar{x}\|_2^2\right)/(2N)}. \tag{7.18}$$

In some cases, the centers overlap exactly. The AP with linear kernel does the worst, missing halves of the rings in the chain link data, and only finding the outer edge of the classes in the Banana dataset. Here, L_2 is not used to build the similarity matrix because it was found to yield similar results to the linear and quadratic polynomial kernels, in that it often identified centers on the extremes of the variable space.

Comparing the number of centers found more exactly across more than just these three problems, results are shown in Table 7.1. Here AP was used on all problems except for the Half-Moon dataset, where LAP and PLAP were used because there are over 14,000 exemplars. Again, in the large-scale AP techniques an initial set of landmark points is found using *maximin* beginning nearest the median and using the L_2 metric. Once that initial set



(a) Banana. (b) Chain. (c) Hepta.

Figure 7.3: Center Comparisons.

is constructed, then the similarity matrix for a specific kernel is input to AP. Three things are of primary interest. First, the linear kernel yields centers that are not as desirable in AP. This is supported by the centers shown in Figure 7.4 for the Half-Moon dataset. The linear kernel, and in fact the quadratic polynomial kernel and L_2 similarity also, generates centers on the extremes of the variable space. Although these shape the boundary of the data, they do not shape the boundaries of objects in the data. Second, Xmeans finds a large number of singleton clusters on the Pima dataset, and overestimates k comparatively. Robust Xmeans, with its multiple starts, avoids this. Finally, the centers found by LAP and PLAP on the Half-Moon dataset are much more useful than the two found by Xmeans for purposes of determining class structure. Again, looking at Figure 7.4, both LAP and PLAP with Gaussian similarity generate centers that find the shape of the two classes. Here, $m = 1,000$ was used. LAP also approximates the borders of the two classes very well. Other kernels for similarity focus on the edges of the data. Meanwhile, Xmeans correctly estimates that there are two groups, but the centers are not of high informational value. The centers from LAP and PLAP on the Chain Links dataset, where $m = 100$, are shown in Figure 7.5. LAP perfectly finds the shape of each ring.

	Banana	Half-Moon	Half-Moon (PLAP)	Chain-Link	Hepta	Iris	Pima
AP - Gaussian	21	35	16	32	9	14	74
AP - InvMultiQuadric	22	36	18	35	11	15	85
AP - Linear	7	9	5	12	6	5	19
Xmeans	7	2	--	69	7	11	158
Robust Xmeans	6	2	--	61	7	14	40
BIC-Means	7	2	--	75	5	10	82

Table 7.1: Center Number Comparisons.

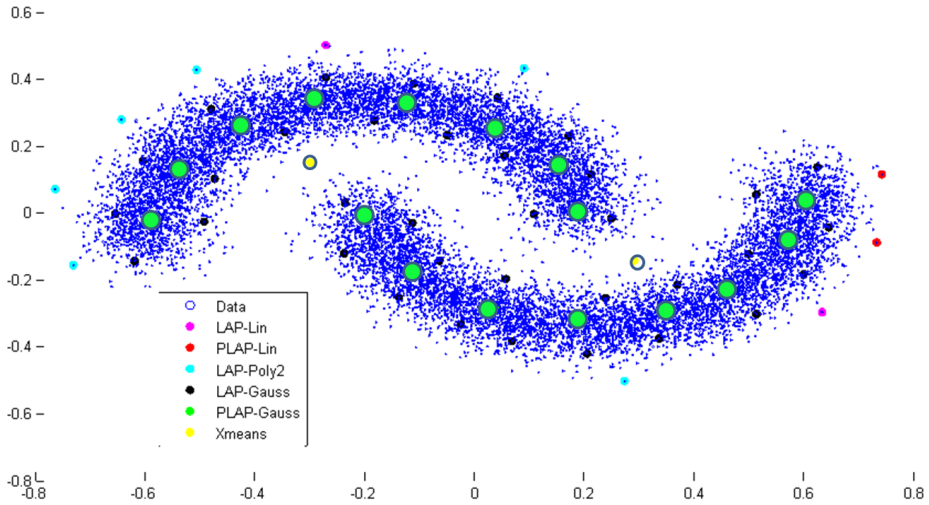


Figure 7.4: Half-Moons Comparison.

For HSI, some representation of what Xmeans and k-means centers correspond to was already analyzed in Section 3.9.1. Unlike their k-means counterparts, the AP centers are actually exemplars from the dataset. Therefore, they are directly interpretable. Figure 7.6 shows the 38 centers found for ARES1D using $m = 1000$ and LAP, overlaid in red onto the RGB image. Here, the Gaussian kernel was again used for the similarity matrix, with σ again set using the average distance to the mean of the data. Pixels from different materials are selected, including a couple of target pixels. Figure 7.7 shows the pixel signatures for

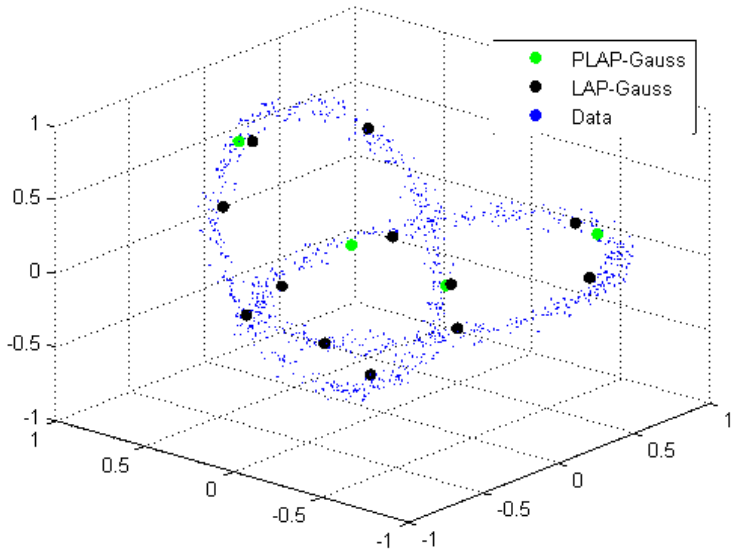


Figure 7.5: Chain Links Landmark Version Comparison.

the 47 centers found for VirginIslands1 using $m = 500$ and LAP. They appear to be evenly spaced spectra, and this is confirmed in Figure 7.8(a). Pixels corresponding to ships, water, and land are all included in the centers. The other plots in Figure 7.8 depict the centers for the cases of $m = 1000$, using the normalized pixel signatures, and PLAP. Using $m = 1000$ instead of $m = 500$ increased the number of centers from 47 to 62. Using normalized data with $m = 100$ increased the numbers of centers from 62 to 86. The PLAP algorithm reduced the candidate centers to a total of only nine pixels. In all cases, ships, water, and land are represented.

LAP is advantageous in comparison to PLAP because it is not random, yields more centers in general, and it is more computationally efficient, scaled linearly according to the number of subsets used in PLAP. PLAP may be more advantageous in that it gives a heavily reduced skeleton. Both methods may be advantageous in comparison to Xmeans because they yield a lower number of centers and use actual exemplars for the centers.

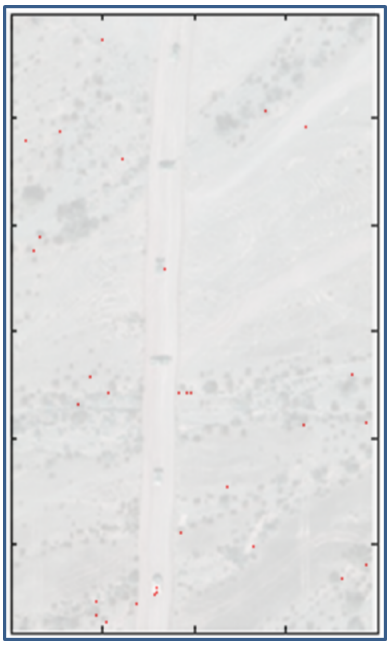


Figure 7.6: ARES1D LAP with $m = 1000$.

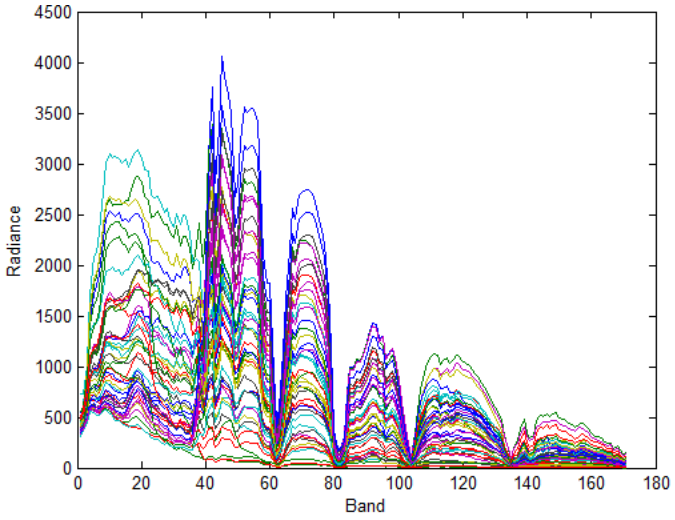
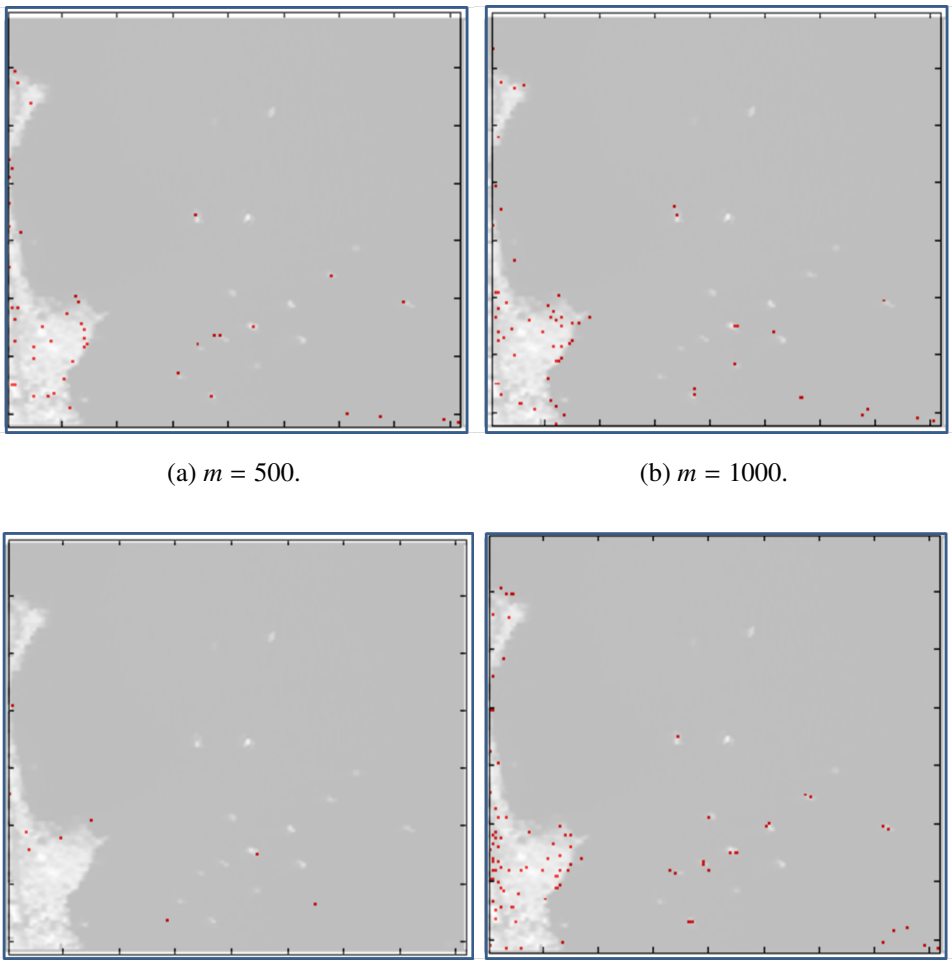


Figure 7.7: VirginIslands1 LAP Centers.



(a) $m = 500$. (b) $m = 1000$.

(c) $m = 1000$, PLAP. (d) $m = 1000$, Normalized Data.

Figure 7.8: VirginIslands1 LAP Center Comparisons.

However, they may be disadvantageous if they include anomalies, as X-means averages the clusters and reduces the negative effect of any anomalies. Table 7.2 shows estimated k and the time to estimate these k centers for various skeleton generating methods for a forest, desert, and water dominated image. For Xmeans, a set of full cluster splits was allowed until 600 clusters were under consideration for splitting. This desired upper bound was chosen because the size of the Gram matrix needs to be limited for KPCA due to memory

and computational constraints. Only the LAP algorithm is truly deterministic, and so the remaining results are means of ten runs where the variation observed was not high. The k-means frameworks, despite their performance enhancements, are still computationally inefficient once k increases to a moderately large value. BIC-means is more efficient, because the whole set of clusters is only re-evaluated after the splitting is done. The bulk of its computational expense is due to the last clustering at the estimated k. In practice, this suggests that it may be more prudent to use a fixed k rather than trying to estimate one, basing this number either on a possible number of materials in the image or on an acceptable size for the resulting Gram matrix.

	VirginIslands1		ARES1F		ARES1D	
	k	Time(s)	k	Time(s)	k	Time(s)
Xmeans	819	246	982	678	1003	1035
Robust Xmeans	904	781	946	1812	988	3359
BIC-means	600	53	600	381	600	863
Robust BIC-means	600	207	600	552	600	1174
LAP (m=500)	47	35	38	40	38	85
PLAP (m=500)	12	162	12	138	13	298
LAP (m=1000)	62	81	59	86	56	171
PLAP (m=1000)	10	274	11	277	7	537

Table 7.2: HSI Mean Center Numbers and Times.

The LAP and PLAP results are interesting because they suggest that a very small number of exemplars may constitute an acceptable skeleton. To further understand how these are being chosen, Figure 7.9 shows the 1000 landmarks from the initial *maximin* skeleton, on which AP is used, overlaid in red onto the natural image. In the VirginIslands1 image, both targets and land are heavily present, while water pixels seem to be less prevalent. Similarly, in ARES1D the landmarks include a large number of target and brush pixels, but also includes road and dirt pixels. This is in contrast to the k-means algorithms, where Figure 7.9(b) and (d) depicts each cluster's pixel index as a color for $k = 64$. The

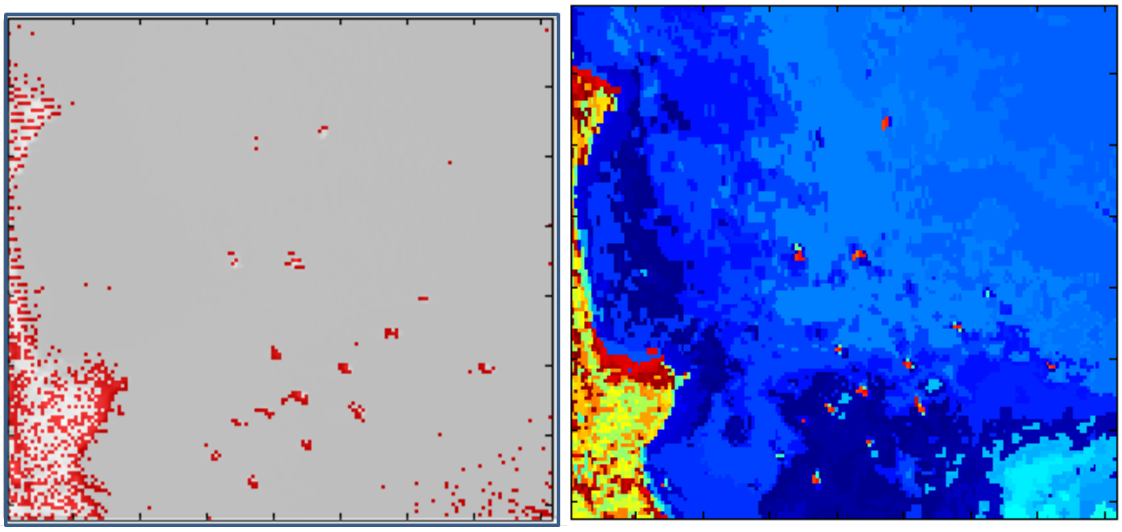
cluster centroids are generally the mean of pixels with the same color. These results suggest that using k-means centroids may be better for constructing background estimates, while AP centers may be better for distinguishing target pixels from background pixels that are more likely to be identified as false positives, assuming the target centers are either removed at some stage or do not negatively affect score maps for this purpose. Further discussion and success of these methods in generating good KPCs is evaluated next.

7.3.3 Resulting KPCs Analysis.

Again, smaller data can be used to help investigate. Figure 7.10 shows the true KPC scores, and NyApprox estimates for the Pima dataset using a Gaussian kernel, the standardized data, and $\sigma = 1.99$. Figure 7.11 shows the scores from small clustering skeletons. In the case of NyApprox, it is clear that as the skeleton gets smaller relative to the size of the data, more error is incurred. However, the scores themselves are still fairly accurate. Meanwhile, the cluster skeletons yield similar behaving scores, but that are much different than the scores from NyApprox. Arguably, this is favorable, as a certain subset of exemplars appear to be different than the rest. This suggests that although these cluster centroids are not as favorable for approximating the true eigenvectors, they may be useful in providing more meaningful mappings.

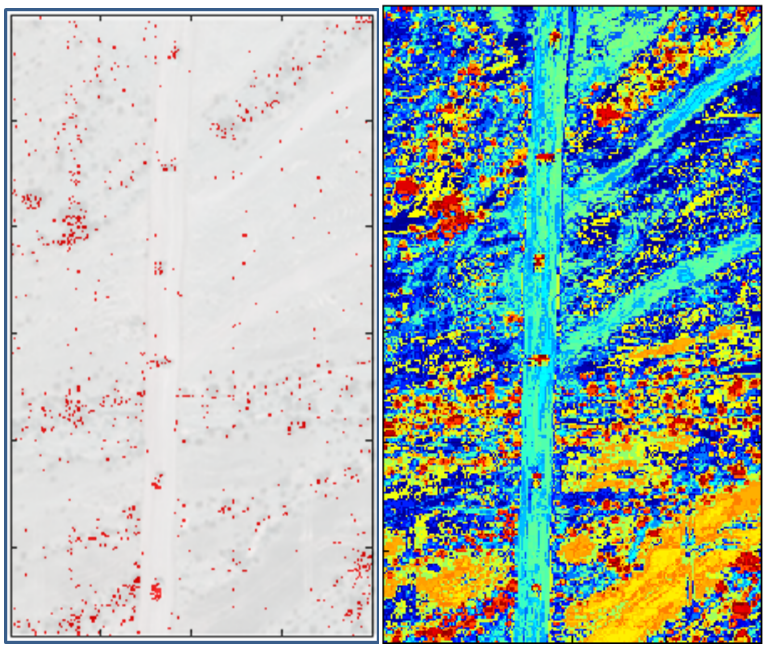
Now, consider the ARES1D image. Figure 7.12 shows sample 'best' mappings from KFA using a Gaussian kernel, and LAP and k-means skeletons. In Figure 7.12(a) and (c), σ was set as a function of the average distance of the data to its mean, while in (b) and (d) it was set as a function of the average distance of the centers to their mean. Thus, $\sigma \leq 4510$ for these maps, and they are shown without any smoothing applied. These mappings all appear to show the anomalies.

Figure 7.13 shows sample maps from KFA using $\sigma = \sqrt{20}$ and using the scaled images. The first row shows maps on which targets are obvious for k-means and LAP



(a) VirginIslands1 Maximin.

(b) VirginIslands1 k-means.

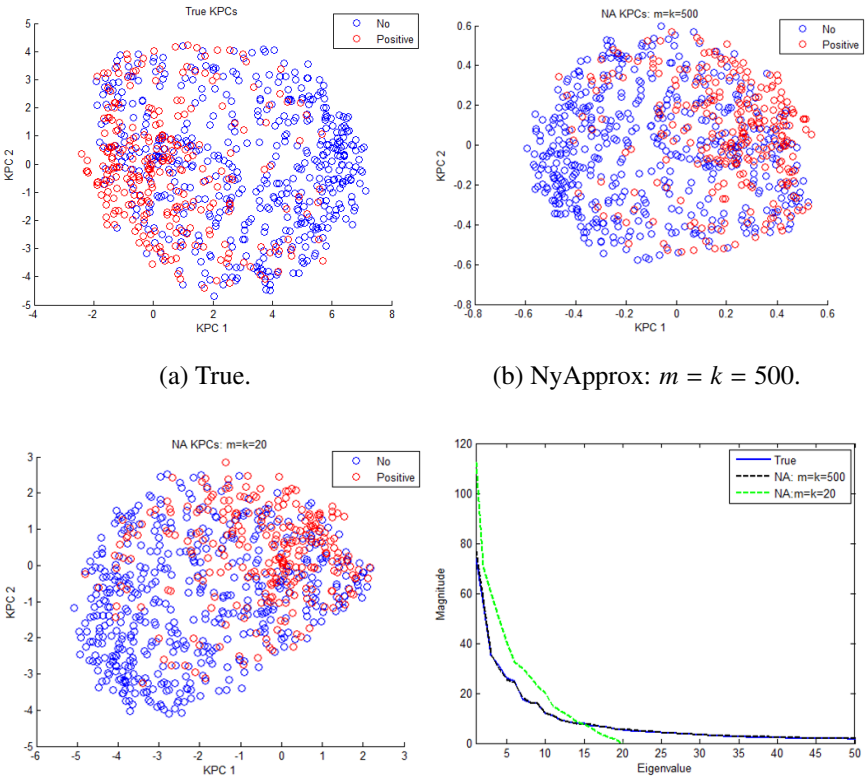


(c) ARES1D Maximin.

(d) ARES1D k-means.

Figure 7.9: Maximin Landmarks & k-means Assignments.

skeletons. The LAP skeleton contained four targets. Therefore, the second row shows corresponding LAP maps where these were removed; note the improvement in score



(a) True.

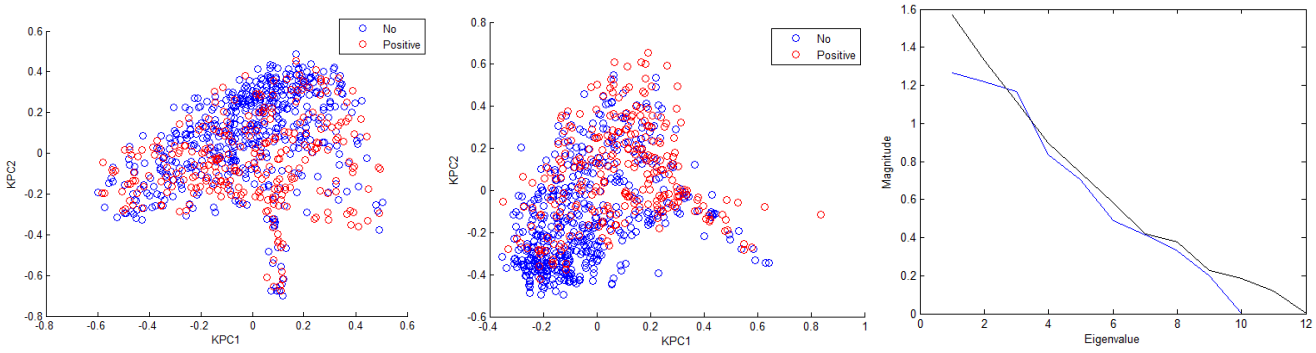
(b) NyApprox: $m = k = 500$.

(c) NyApprox: $m = k = 20$.

(d) Eigenvalues.

Figure 7.10: Pima Eigenvectors and Values.

magnitudes and noise. After much experimentation, a few things were clear. The data skeletons are sensitive to outliers, as evidenced here. The choice of σ and whether or not to scale the data also has a big impact on the quality of the maps. Figure 7.14 further exemplifies this, where these maps correspond to ARES1F. The KPC scores for ARES1F, here constructed using NyApprox with $m = 600$ and $r = 500$, do not cleanly break out the target class. Instead, the road and other materials also appear potentially anomalous. Applying FA, certain targets appear on different maps, but not strongly. Using KPCA and KFA with a much larger spread, the targets become more obvious, but the background is

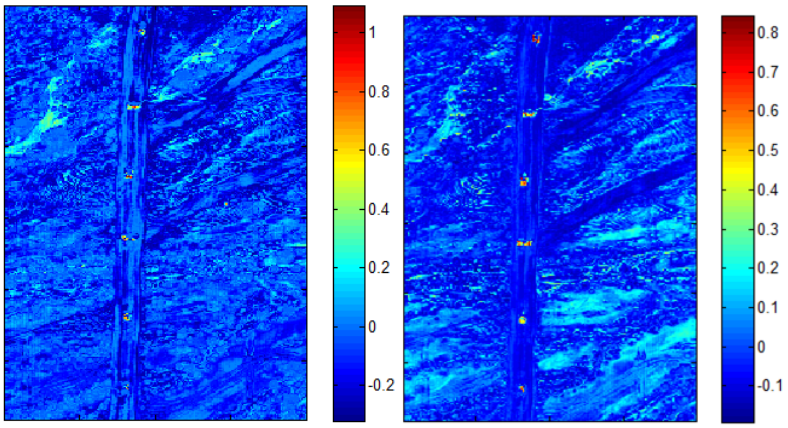


(a) k-means: $k = 10$. (b) AP: 12 Centers. (c) Eigenvalues.

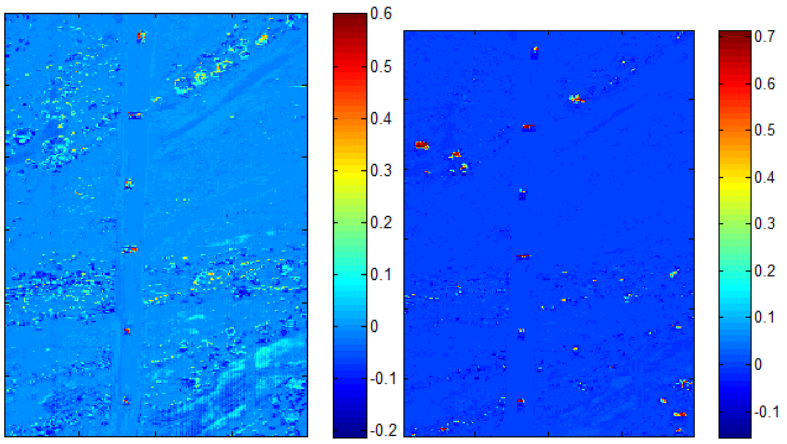
Figure 7.11: Pima Eigenvectors and Values.

not as clean on the maps. However, reducing the spread again and using a relatively small k-means provides cleaner maps where the targets appear more obvious.

Other than to say that the choice of σ, kernel, and skeleton size greatly impact the quality of the mappings, it is very difficult to quantify how these affect the scores in a general sense, accurately. A KPC or factor with high scores is not necessarily a good component for finding anomalies. Similarly, a component with high SNR is not always the best. The n-dimensional technique from Chapter 5 can be used in an effort to show the benefit of different skeletons and size. For example, Figure 7.15 depicts resulting factor scores for k-means skeletons with MDSL applied after the KPCA. It can be seen that more separation is achieved between anomaly and background with smaller skeleton size. This is because, in part, there are fewer maps to rotate. Figure 7.16 show skeletons in a similar fashion for ARES1F. Again, smaller is better, and k-means appears to give a cleaner set of maps than NyApprox. Further analysis revealed $k = 50$ to be psuedo-optimal for skeleton size, as fewer centroids did not generate enough quality maps, while more



(a) LAP: $k = 56$, Data Mean. (b) LAP: $k = 56$, Center Mean.

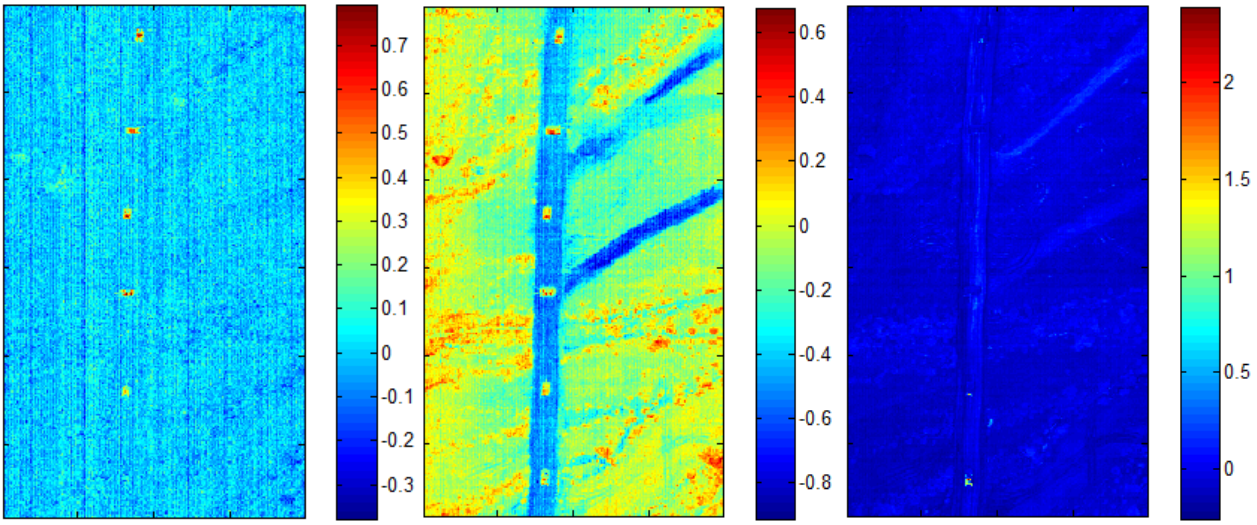


(c) k-means: $k = 500$, Data Mean. (d) k-means: $k = 500$, Center Mean.

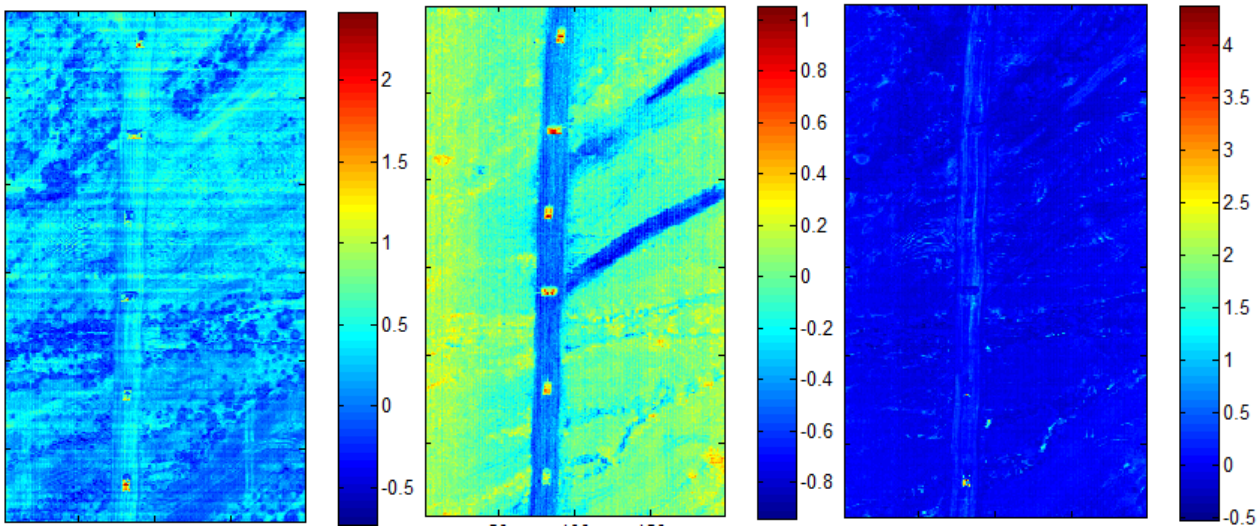
Figure 7.12: ARES1D KFA Scores.

enabled anomalies to spread across maps and be more difficult to detect. This is discussed further shortly, during analysis of the final algorithm.

Regardless of the difficulty of quantification, it has been shown that desirable mappings are possible using the non-linear techniques. Further, if k-means is used to build the skeleton with a moderate k (*i.e.*, < 250), it is, relatively, computationally inexpensive or equivalent in comparison to AP and NyApprox. Any form of AP is problematic due to the sensitivity of the mappings to outliers, and the possibility that AP uses an anomaly in the



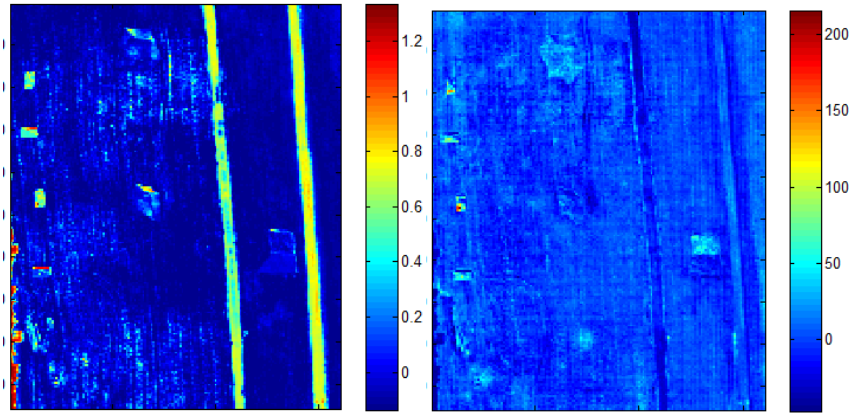
(a) k-means: $k = 500$. (b) LAP: $m = 1000$. (c) LAP: $m = 1000$.



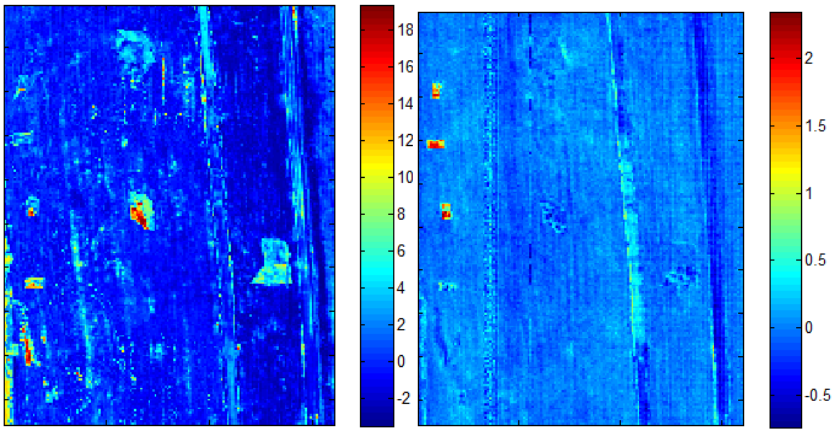
(d) LAP: $m = 1000$ (No Targets). (e) LAP: $m = 1000$ (No Targets). (f) LAP: $m = 1000$ (No Targets).

Figure 7.13: ARES1D KFA Scores: $\sigma = \sqrt{20}$.

skeleton. Meanwhile, k-means only uses an average of anomalies in the worst-case, thus slightly mitigating the outlier sensitivity. Table 7.3 shows run-times for the various skeleton



(a) PC $\sigma = \sqrt{20}$. (b) FA $\sigma = \sqrt{20}$.



(c) FA $\sigma = 5207$. (d) k-means: $k = 100$, $\sigma = \sqrt{20}$.

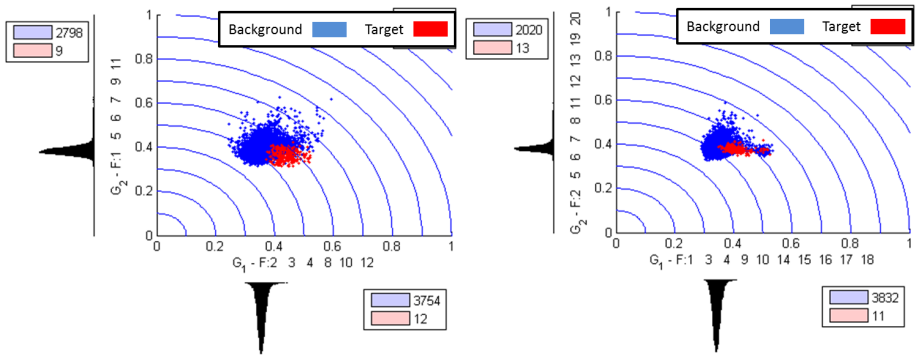
Figure 7.14: ARES1F Scores.

generation techniques, with and without factor analysis applied to the kernel components, where statistics were taken over ten values of σ ranging from 21 to 8000.

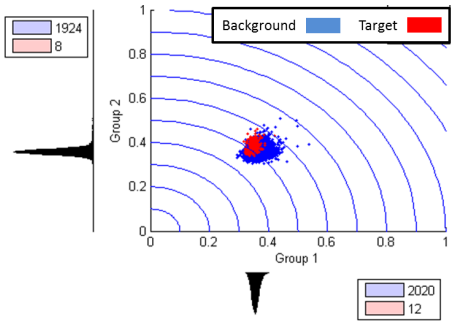
In the experimentation of these mappings and factors, a few things became clear. Using the scaled data and smaller σ was generally a better approach. In Chapter 8, larger σ perform better, but this is due to a difference in the algorithms. There, finding a boundary is important, while here, generating maps that show the anomalies is important. The Gaussian

	Mean	StDev
NyApprox (r=50, m=100)	17.64	2.07
NyApprox FA (r=50, m=100)	20.33	1.41
NyApprox (r=100, m=150)	17.38	1.79
NyApprox FA (r=100, m=150)	27.44	6.44
NyApprox (r=500, m=600)	17.76	2.53
NyApprox FA (r=500, m=600)	29.18	5.24
Kmeans (k=50)	21.11	3.51
Kmeans FA (k=50)	24.88	5.66
Kmeans (k=100)	54.81	21.15
Kmeans FA (k=100)	82.82	92.23
Kmeans (k=500)	522.71	49.28
Kmeans FA (k=500)	565.60	67.05
LAP (m=250)	20.79	1.07
LAP FA (m=250)	20.77	1.06
LAP (m=1000)	92.19	10.82
LAP FA (m=1000)	99.45	29.71
PLAP (m=250)	191.89	193.00
PLAP FA (m=250)	188.38	186.43
PLAP (m=1000)	546.33	585.37
PLAP FA (m=1000)	553.18	561.96

Table 7.3: Skeleton Generation Times(s).



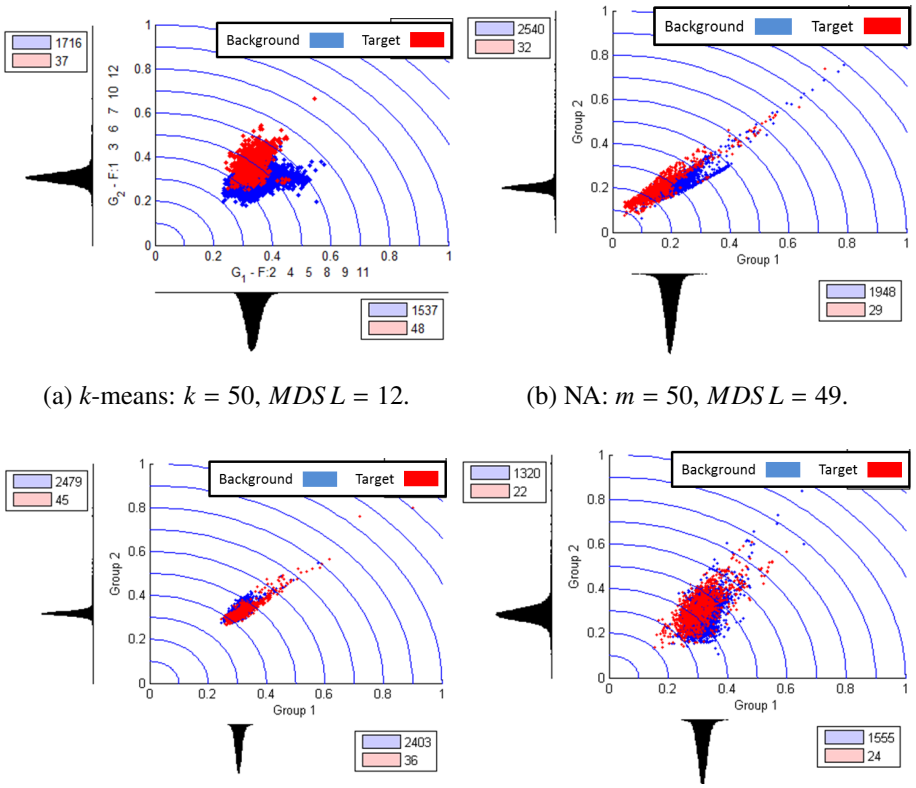
(a) $k = 50$, $MDSL = 12$. (b) $k = 100$, $MDSL = 20$.



(c) $k = 500$, $MDSL = 36$.

Figure 7.15: ARES1D k-Means Skeleton Comparisons.

kernel was also the best performer, where the polynomial kernel never provided maps with information not found by the Gaussian. NyApprox and k-means also seemed to provide the best skeletons, most consistently. This was because AP directly allows outliers as landmark points; although NyApprox can as well, in that algorithm eigenvectors are approximated across all exemplars. Applying approximate factor analysis was also necessary in order to consistently provide better score maps. It is clear that non-linear mappings can provide components that reveal anomalies well, but that parameter and sub-algorithm choices greatly affect success. This also leads to the conclusion that it may be difficult to standardize settings of an algorithm across images.



(a) k-means: $k = 50$, $MDSL = 12$.

(b) NA: $m = 50$, $MDSL = 49$.

(c) NA: $m = 100$, $MDSL = 94$.

(d) NA: $m = 500$, $MDSL = 98$.

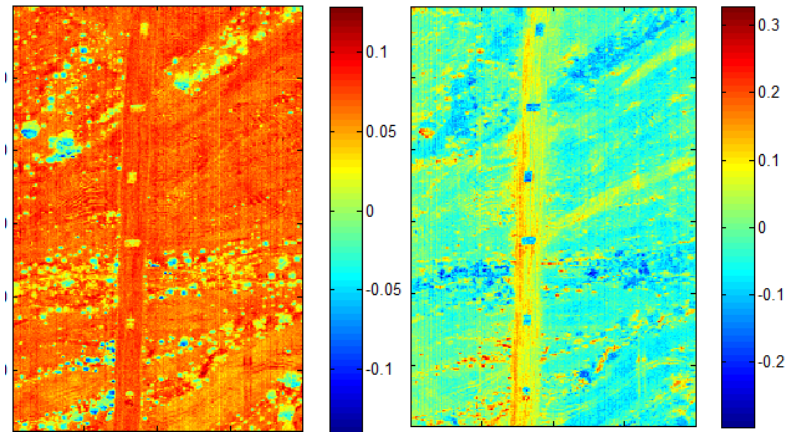
Figure 7.16: ARES1F k-means and NyAppox (NA) Skeleton Comparisons.

Before proceeding, recall from Section 3.5 that Local Linear Embedding (LLE) is a special form of KPCA, where it is assumed that the data lies on a manifold and local linear reconstruction based on neighbors is used to produce the lower-dimensional embedding. Therefore, it also warrants consideration because it uses similarity and manifold information. Given the ARES1D image, LAP produces 56 centers. Using these centers as training, or the centers from k-means with $k = 56$ as training, the resulting LLE embeddings are highly noisy. This, and similar results on other images suggests that LAP produces too few landmarks to provide a good approximation with LLE. Figure 7.17 shows specific embeddings generated using a training set of 250 and 500 centroids

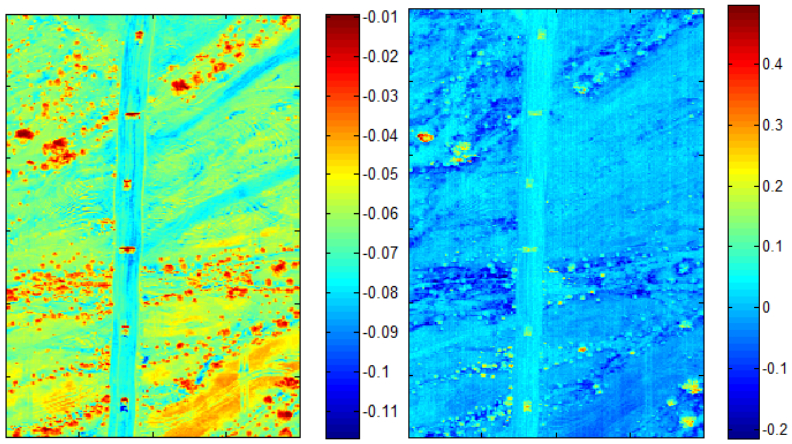
from k-means with those respective k's, various numbers of nearest-neighbors for the reconstruction, and with/without using robust LLE. The numbers in parentheses in the caption denote the embedding number in terms of smallest eigenvalue magnitude. The most useful embedding is that based on the largest number of centroids and nearest neighbors. In general, increasing the number of centroids for the training set improved the embeddings on different images. Unfortunately, larger skeletons and nearest-neighbors greatly increase the computational expense for LLE. Additionally, the mapping in Figure 7.17 is still not as clean as those from factor analysis in Chapter 6, as the brush has higher scores than the true anomalous pixels. Therefore, LLE is likely not appropriate here, and so next the use of KPCA and KFA is investigated further within the IGFAAD framework.

7.4 KIGFAAD

The IGFAAD framework from Chapter 7 can be extended easily to the kernel case, as the mappings from the skeleton seem to behave in a similar fashion, although anomalies spread more widely across the factors and they are not as interpretable (due to being loaded on landmarks). The modified framework is shown in Figure 7.18. Here, the only changes to the algorithm framework are that the data is scaled by its maximum value in the first step, a data skeleton is used for the eigen-analysis, and that the covariance eigen-decomposition and subsequent factor analysis occurs in the kernel space. For a skeleton choice, consider that NyApprox is still very much random, and can have singularity issues in practice. k-means is random because of its initial solution, but the refined start reduces the randomness. For moderate k, k-means and NyApprox have a similar computational expense. Additionally, k-means essentially smooths the skeleton by using the centroids instead of exemplars, and so k-means is more resistant to outliers. Therefore, it is chosen to generate the skeletons. The LAP algorithm was also tested, using $m = 1000$, but did not



(a) $k = 250$, $knn = 5$, Robust (1). (b) $k = 250$, $knn = 5$, Robust (3).



(c) $k = 250$, $knn = 10$, (3). (d) $k = 500$, $knn = 10$, (3).

Figure 7.17: LLE Scores.

perform well in general and showed an influence by outliers, often having at least a few in the skeleton that weakened results.

Kernel RX (KRX) is initially the best comparison for any kernel-based algorithm, as its global version has been shown to perform very well on certain HYDICE images [133]. In their work, Kwon and Nasrabadi [133] used $\sigma = \sqrt{20}$ for the Gaussian kernel, and a $k = 600$ k-means centroids training matrix for KRX on an image very similar to ARES1D.

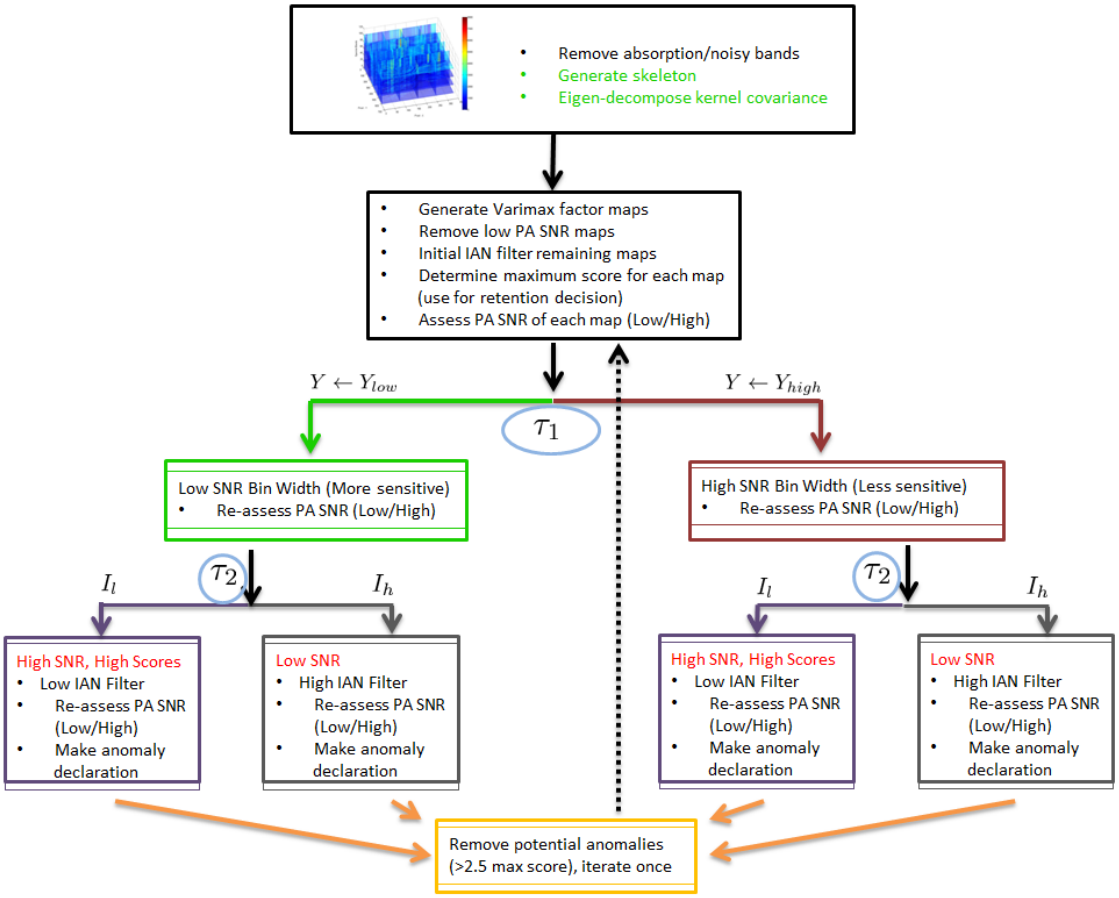


Figure 7.18: KIGFAAD Process.

In experimentation of the images in this research, that value for σ also appears to be psuedo-optimal, while $k < 600$ is still suitable. Figure 7.19 shows the ROC curves for the basic seven problem test set used throughout this research, using $k = 200$ for k-means. It can be seen that KRX performs fairly well overall, but better on some problems than others.

Also shown in Figure 7.19 are initial guesses at operating points for KIGFAAD, where they were set according to the settings in Table 7.4, and with a k-means skeleton for $k = 50$. In practice, having at least 20 to 30 factors available seemed to provide reasonable maps, and so $k = 50$ was chosen initially to ensure that number of maps and to keep the clustering

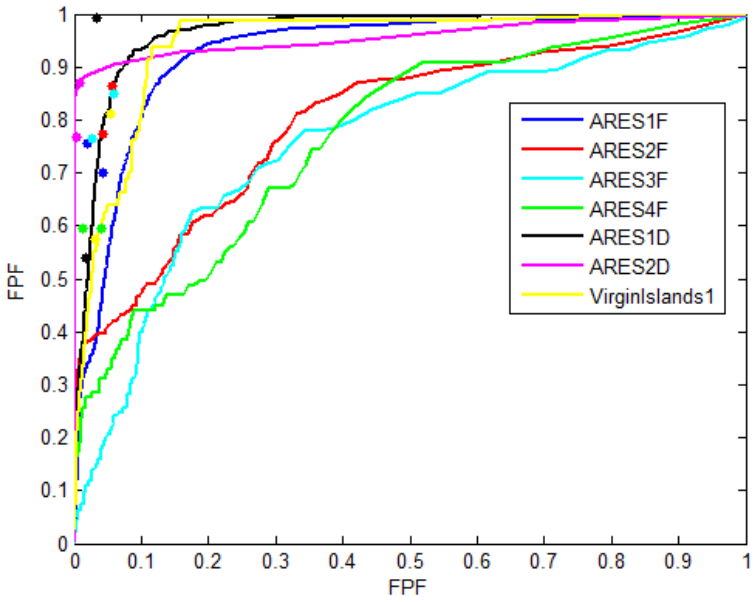


Figure 7.19: KRX ROCs vs. Initial KIGFAAD Operating Points.

efficient. $\sigma = \sqrt{20}$ was still used for the Gaussian kernel on the scaled data. Actual TPF and FPF values are shown in Table 7.5. Here, previous settings were taken from IGFAAD, except that the score magnitude thresholds were reduced, the pixels per bin values were reduced, and the dynamic factor adjustments were eliminated by using the same bin widths at all stages. The score thresholds and pixels per bin have to be reduced here because the kernel factor maps are different in nature than their linear counterparts. In general, they have lower scores and lower SNR. Given, that several parameter values were copied from the linear case, and that the dynamic aspects of the algorithm were essentially turned off, the results also indicate that these initial settings are far from optimal. Interestingly, the settings are still on or to the left of the KRX ROC curves, which is very promising. The ARES1D results in particular are already at an extremely good performance level. Additionally, simply due to the nature of clustering with $k = 50$ versus $k = 200$, the algorithm is more

efficient than KRX. In addition to indicating competitiveness with KRX, this indicates that KIGFAAD is advantageous over KRX because an actual operating point can be found.

Table 7.4: KIGFAAD Experiment Settings.

Parameter	Name	Initial	Experiment
t_{MS}	Max Score Threshold	0.5	0.1,0.55,1
t_{SNR}	Bin Width PA SNR Threshold	-1	-3,1,1
$I_{initial}$	Initial IAN Iterations	4	0,3,6
I_h	IAN Iterations High	20	0 20 40
I_l	IAN Iterations Low	12	0 10 20
$Y_{initial}$	Pixels Per Bin Initial	100,200	50,150,250
Y_{low}	Pixels Per Bin Low	100,200	24,74,124
Y_{high}	Pixels Per Bin High	100,200	100,150,200
τ_1	Bin Choice SNR Threshold	(τ) 7.17	3 7 11
τ_2	Smoothing Choice SNR Threshold	10	5 10 15
t_s	Score Magnitude Threshold	10	1,4,7

	Y=100		Y=200	
	TPF	FPF	TPF	FPF
ARES1F	0.7001	0.0427	0.7547	0.0189
ARES2F	0.8632	0.0558	0.7720	0.0420
ARES3F	0.8482	0.0590	0.7655	0.0273
ARES4F	0.5963	0.0393	0.5963	0.0116
ARES1D	0.9915	0.0331	0.5404	0.0182
ARES2D	0.8700	0.0082	0.7686	0.0027
VirginIslands1	0.8125	0.0549	0.5750	0.0311

Table 7.5: Initial KIGFAAD Results.

To better optimize the KIGFAAD algorithm, a similar approach to that used in Chapter 6 is taken here. For an RSM experiment with which to optimize the KIGFAAD algorithm, experiments using three-level full-factorials over the settings shown in Table 7.4 were performed. After adjusting ranges and optimizing, the settings in Table 7.6 were found. This provided competitive TPF and FPF rates to the linear algorithm, however, these experiments were done over a fixed instance of the robust clustering for each image. Upon repetition, it was discovered that even small changes to the skeleton (one different centroid, for example) could greatly impact results. Changes in the TPF up to 0.2 occurred. This does make sense, as the skeleton is very small, and approximates the kernel factors for all N pixels. Therefore, any skeleton with randomness, even if minimal, is not desirable.

Table 7.6: KIGFAAD Optimal Settings.

Parameter	Name	Optimal
t_{MS}	Max Score Threshold	0.55
t_{SNR}	Bin Width PA SNR Threshold	1
$I_{initial}$	Initial IAN Iterations	5
I_h	IAN Iterations High	32
I_l	IAN Iterations Low	6
$Y_{initial}$	Pixels Per Bin Initial	123
Y_{low}	Pixels Per Bin Low	111
Y_{high}	Pixels Per Bin High	120
τ_1	Bin Choice SNR Threshold	6.5
τ_2	Smoothing Choice SNR Threshold	13.5
t_s	Score Magnitude Threshold	6

In order to resolve this issue, the deterministic method developed by Su and Dy [197] was used. They found that using PCs of the data could be used to yield a low distortion initial guess for centroids. In particular, the data is iteratively split into clusters by choosing the cluster at a given iteration with maximum distortion, projecting the cluster's membership onto its first major principal component, and then splitting the cluster's membership into two new clusters according to which side of the component's mean it falls on. This is beneficial because it is efficient, deterministic, and has low distortion because the principal components are already sum square error optimal in terms of reconstruction.

Using this to build the skeleton, k, the iteration score threshold, using MDSL or not as an eigenvector cut-off, σ, the number of iterations, using BACON as a pre-screen to remove anomalies from the skeleton, and the general algorithm parameters were all tested using the RSM framework again in a very large series of experiments. Removing eigenvectors before the rotation with MDSL generated too few maps for $k \leq 50$ on many of the problems, and even did so for a few problems with k up to 200. One example was ARES1F, where the TPF dropped to 0.507 with only 34 retained maps of the possible 200. Changing the iteration criterion as a function of t_{MS}, *i.e.*, deviating from the constant 2.5 in the criterion, as well as increasing iteration led to increased false positives on some problems. Although that occurred, leaving the iteration as it was in IGFAAD did generally improve results, though it simultaneously decreased TPF and FPF rates on certain problems. Using BACON as a form of pre-screen made the centroids too clean, and led to increased false positives. In general, decreasing k from a 45-50 range did not allow enough variation into the skeleton to model the background properly, such that TPFs reduced. Similarly, increasing k allowed pixels nearer the outliers into the skeleton on some problems, and reduced TPFs or increased FPFs. With larger k, more maps (factor score sets) are rotated, such that the resulting maps provide less meaning. In fact, the range of $k = 45$ to 50 seemed to balance the trade-off between not allowing outliers to overly influence the skeleton, with providing enough

maps for discrimination and few enough such that the rotation would yield few good maps. The rotation itself was not a problem, as testing the KPCs without factor analysis applied yielded maps of lesser quality. It must also be remembered that the loadings for kernel FA are on the skeleton exemplars, thus making selection of the skeleton so important. σ also had some slight impact, but again, interacted with other parameters. σ was varied from $\sqrt{10}$ to $\sqrt{35}$.

The best general parameters based on RSM experiments turned out to the same as those shown already in Table 7.6, as those particular skeletons were very similar to the deterministic version. Results from a few different investigations and kernel algorithm-specific parameter settings are shown in Table 7.7. Here, the headers denote the number of centroids, whether or not iteration was performed, and σ, in that order. Each row is also conditionally formatted by magnitude, where green reflects 'best' and red reflects 'worst' performance. It is clear that the non-linear algorithm is more sensitive to changes in image characteristics than the linear, and that certain settings for one image are not optimal for another. However, with $k = 50$, one iteration, and $\sigma = 5$, the algorithm is generally and most competitive, yielding promise that further investigation could lead to a version that more consistently outperforms the linear.

KIGFAAD, in algorithmic form is given as Algorithm 7.2. A final comparison of this algorithm at its new settings to the published KRX is shown as Figure 7.20. As shown, KIGFAAD operates at a much better point in the TPF/FPF space.

KIGFAAD results on the remaining images with truth masks are given in Table 7.8. It has higher detection, but higher false positives on run03m20, and does arguably better than IGFAAD on 4Ships2 and Scene1. Meanwhile, the skeleton is highly inaccurate for the Ship1 and HyMAP images. In the case of HyMAP, there are only 145 target pixels, and so the skeleton is not representative enough of the entire background to prevent the high false positives. In the case of Ship1, the scene is dominated by water but also has land.

Algorithm 7.2 KIGFAAD Algorithm

1: $X^s_{N\times p} \leftarrow X_{N\times p} / \left(\max_{1\le i\le N, 1\le j\le p} x_{ij} \right)$: data is scaled by its maximum value.

2: Generate data skeleton $S_{k\times p}$ with k centroids using deterministic k-means on $X^s_{N\times p}$.

3: Find kernel eigenvectors V and eigenvalues Λ of $\hat{K}_{k\times k}$ formed on $S_{k\times p}$.

4: Perform approximate kernel factor analysis with Varimax rotation on the kernel factor loadings $\Lambda^{1/2}V$. Compute the unweighted least squares estimate scores $F_{N\times k}$.

5: **if** $|\min(F^i)| > \max(F^i)$ for $1 \le i \le k$, **then** $F^i \leftarrow -F^i$, **end if**.

6: $snr_i \leftarrow PA\ SNR(F^i)$ with bin width $Y_{initial}/N$ using first zero-bin histogram.

7: **if** $snr_i > t_{SNR}$, **then** retain F^i, **end if**.

8: $F^i \leftarrow IAN(F^i)$, with $I_{initial}$ iterations. For each factor mapping F^i, $m_i \leftarrow \max(F^i)$.

9: Retain any factor mapping with $m_i \ge t_{MS}$. If none satisfy this, declare no anomalies and stop. Otherwise, go to Step 10.

10: $snr_i \leftarrow PA\ SNR(F^i)$ with bin width $Y_{initial}/N$ using first zero-bin histogram.

11: **if** $snr_i \le \tau_1$, **then** $Y^i \leftarrow Y_{low}$ and $snr_i \leftarrow PA\ SNR(F^i)$ with bin width Y_{low}/N; **else** $Y^i \leftarrow Y_{high}$ and $snr_i \leftarrow PA\ SNR(F^i)$ with bin width Y_{high}/N, **end if**.

12: **if** $snr_i \ge \tau_2$ & $m_i \ge t_s$, **then** $F^i \leftarrow IAN(F^i)$, with I_l iterations; **else if** $snr_i \le \tau_2$, **then** $F^i \leftarrow IAN(F^i)$, with I_h iterations, **end if**.

13: Repeat Steps 10-11 using Y^i as the initial bin width.

14: Define $\eta_i \leftarrow PA\ SNR(F^i)$ as the threshold from the first zero-bin histogram, using Y^i.

15: If first iteration, remove any pixel j with $F^i_j > 2.5 \times t_{MS}$ from data used for covariance estimate and go to Step 2. If no such pixels exist, or second iteration, go to Step 16.

16: **if** $F^i_j > \eta_i$, **then** declare pixel j anomalous, **end if**.

		IGFAAD	k=50, No Iter, sqrt(15)	MDSL 1Iter2.5	k=50, Iter, 5	k=50, Iter sqrt(15)	k=45, Iter, sqrt(15)	k=45, No Iter, sqrt(15)
ARES1F	TPF	0.9652	0.9434	0.3466	0.9643	0.9563	0.9394	0.9364
	FPF	0.0178	0.0513	0.0265	0.0313	0.0640	0.0609	0.0312
ARES2F	TPF	0.9642	0.9544	0.7362	0.9772	0.9837	0.9707	0.9609
	FPF	0.0242	0.1047	0.0699	0.0829	0.1070	0.1199	0.0748
ARES3F	TPF	0.8552	0.9310	0.9034	0.9172	0.9310	0.9310	0.9241
	FPF	0.0700	0.0512	0.0324	0.0416	0.0538	0.0462	0.0302
ARES4F	TPF	0.8349	0.8991	0.6514	0.8073	0.8257	0.8165	0.7798
	FPF	0.0334	0.2356	0.0255	0.0286	0.0398	0.0452	0.0245
ARES1D	TPF	0.9489	0.9830	0.8766	0.8255	0.8511	0.9872	0.7277
	FPF	0.0205	0.0366	0.0280	0.0296	0.0347	0.0284	0.0185
ARES2D	TPF	0.9446	0.9847	0.8872	0.8910	0.9006	0.9216	0.9120
	FPF	0.0140	0.0195	0.0064	0.0126	0.0146	0.0109	0.0044
Virgin1	TPF	0.9750	0.9250	0.9000	0.9625	0.9125	0.9625	0.9000
	FPF	0.0724	0.0918	0.0433	0.0680	0.0762	0.0973	0.0746

Table 7.7: KIGFAAD Results.

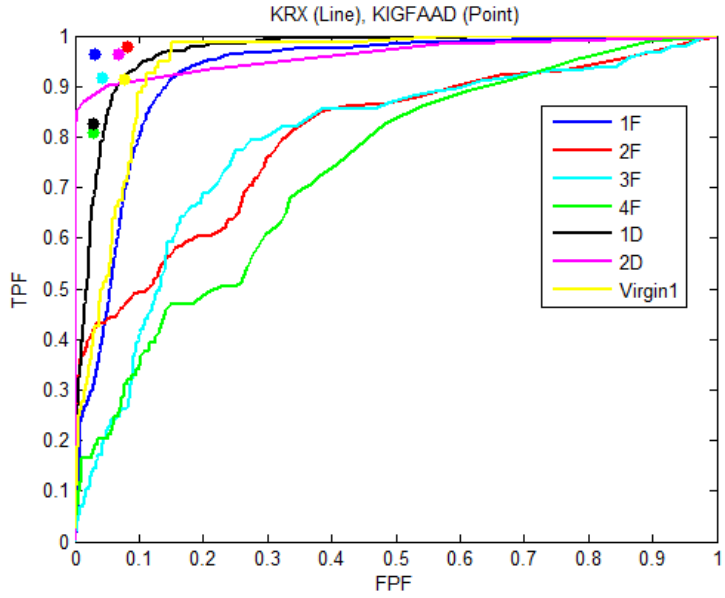


Figure 7.20: KRX ROCs vs. KIGFAAD Operating Points.

The centroids end up being more representative of the water pixels, but there are some that are heavily influenced by land and ships. This, at the current parameter settings, skews

the skeleton so as to provide very few useful maps. Although the other AVIRIS images have land, water, and ships, the exact cluster size here, coupled with the factor rotation, is a combination that provides a less useful skeleton and results in loading too heavily on centroids that do not fully represent any of the classes.

Unfortunately, the nature of the algorithm and its many interactions amongst steps and parameters make it difficult to realize a way to perform optimally across all images, although performing well across most is certainly possible. Nonetheless, the results are still promising, both due to the results on other images, and because these high FPFs did not always occur with different skeletons and/or parameter settings. One way to mitigate this would be to automatically change parameter settings if such a high TPF and FPF were found during the course of the algorithm. A true benefit of the non-linear algorithm is that it is not as susceptible as its linear counterpart to declaring anomalies because of sensor error in a few bands. The use of pixels, rather than bands, to build the covariance helps to avoid this.

		IGFAAD	KIGFAAD
ARES1C	TPF	--	--
	FPF	0	0.0298
ARES2C	TPF	--	--
	FPF	0.0125	0.0405
run03m20	TPF	0.4339	0.5564
	FPF	0.0392	0.0860
4Ship2	TPF	1	0.9789
	FPF	0.1081	0.0568
Scene1	TPF	0.9899	0.8512
	FPF	0.0365	0.0106
Ship1	TPF	0.9112	1
	FPF	0.0646	0.4496
HyMAP	TPF	0.8759	1
	FPF	0.0833	0.7383

Table 7.8: KIGFAAD Results on Other Images.

Next, use of kernel methods within the context of an entirely different framework is discussed. There, anomaly detection is formulated as a boundary problem, where the focus becomes to estimate boundary exemplars in the higher-dimensional space. Data skeletons and kernel selection are still very relevant in this alternative framework.

VIII. Support Vector Data Description

8.1 Literature Review

Support Vector Data Description (SVDD) assumes that non-linear mapped data in a higher-dimensional space can be separated using a hypersphere. This is not in conflict with the consideration that hyperspectral data may not take the shape of an elongated hyperellipsoid, as that observation is largely based on PCA with a Gaussian assumption in the originating space, and not the non-linear mapped space [137]. SVDD is a supervised method, but it is desirable to develop an unsupervised version as class information is not always known.

8.1.1 SVDD for Anomaly Detection.

Banerjee, Burlina, and Diehl [22] extended SVDD for use as an anomaly detector in part using the kernel trick. In order to remove the homogeneous, Gaussian background assumption that is made by detectors such as RX (and the accompanying high false-alarm rates due to multiple terrain classes breaking the Normal assumption), their SVDD detector incorporated a nonparametric background model. The ultimate goal is to estimate the shape and size of the support region for the background. Using Support Vector Machines (SVMs) fewer training samples are needed to accurately characterize the background, no parametric assumption is made in the feature space, over-fitting is avoided, and the support of nontrivial multi-modal distributions can be modeled.

The non-linear SVDD maps the data from the input space to a higher-dimensional feature space using a mapping $\Phi(\boldsymbol{x})$, and models the support of the distribution as a minimum enclosing hypersphere in the feature space. This can add flexibility that the hypersphere based on the original input space does not provide. The resulting hypersphere corresponds to a tighter boundary for the support region in the original input space.

Sampling a set of N exemplars $\mathbf{x}$ to use as background, SVDD attempts to determine the smallest hypersphere $S(R,c) = \left\{\Phi(\mathbf{x}) : \|\Phi(\mathbf{x}) - c\|^2 < R^2\right\}$ that contains the set of mapped training exemplars. This is obtained by solving,

$$\min (R) \text{ subject to } \Phi(\boldsymbol{x}_i) \in S\,,\, i = 1, ..., N. \tag{8.1}$$

The objective can be equivalently replaced with R^2. At first glance this problem seems easy to solve. However, c is unknown and in general, c cannot be estimated directly from the exemplars $\boldsymbol{x}_i$. As it turns out, the radius R and center c can be determined by maximizing the infimum of the Lagrangian dual with respect to the Lagrangian multipliers α_i (expanding the hypersphere norm),

$$L(R, c, \alpha_i) = R^2 - \sum_i \alpha_i \left\{R^2 - (\langle\Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_i)\rangle - 2\langle c, \Phi(\boldsymbol{x}_i)\rangle + \langle c, c\rangle)\right\}. \tag{8.2}$$

To satisfy the Karush-Kuhn-Tucker (KKT) conditions for an optimal solution $\left(\hat{R}, \hat{c}, \hat{\alpha}_i\right)$,

$$\frac{\partial L}{\partial R} = 0 = 2\hat{R} - 2\hat{R}\sum_i \hat{\alpha}_i \Rightarrow \sum_i \hat{\alpha}_i = 1, \tag{8.3}$$

and

$$\frac{\partial L}{\partial c} = 0 = -\sum_i \hat{\alpha}_i\,(2\Phi(\boldsymbol{x}_i) - 2\hat{c}) \Rightarrow \hat{c} = \sum_i \hat{\alpha}_i \Phi(\boldsymbol{x}_i). \text{ (using 8.3)} \tag{8.4}$$

Using these necessary optimality conditions back in L yields,

$$L = \sum_i \alpha_i \langle\Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_i)\rangle - \sum_i \sum_j \alpha_i \alpha_j \langle\Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j)\rangle, \tag{8.5}$$

with $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$. Incorporating the Kernel trick to evaluate the inner products, where k denotes the kernel function, this becomes,

$$L = \sum_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_i \sum_j \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{8.6}$$

with $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$. Or, letting $K_{ij} = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, this is equivalently,

$$\begin{aligned} &\max \inf \qquad L = \sum_i \alpha_i K_{ii} - \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \\ &\text{subject to} \qquad \sum_i \alpha_i = 1. \end{aligned} \tag{8.7}$$

The decision rule to detect an anomaly (or target class) for a test exemplar $\boldsymbol{y}$ is to see if it falls outside the hypersphere (which now represents the background or training class). Plugging in c and k for the dot products, this yields,

$$\begin{gathered} SVDD(\boldsymbol{y}) = \|\Phi(\boldsymbol{y}) - c\|^2 \geq R^2 \Rightarrow \\ SVDD(\boldsymbol{y}) = k(\boldsymbol{y},\boldsymbol{y}) - 2\sum_i \alpha_i k(\boldsymbol{y},\boldsymbol{x}_i) + \sum_i \sum_j \alpha_i \alpha_j k(\boldsymbol{x}_i,\boldsymbol{x}_j) \geq R^2. \end{gathered} \tag{8.8}$$

Use of the Gaussian Radial Basis Function (RBF), $k(\boldsymbol{x},\boldsymbol{y}) = exp\left(\frac{-\|\boldsymbol{x}-\boldsymbol{y}\|^2}{\sigma^2}\right)$ simplifies L to,

$$L = 1 - \sum_i \sum_j \alpha_i \alpha_j k(\boldsymbol{x}_i,\boldsymbol{x}_j), \tag{8.9}$$

and the decision rule to,

$$SVDD(\boldsymbol{y}) = 1 - 2\sum_i \alpha_i k(\boldsymbol{y},\boldsymbol{x}_i) + \sum_i \sum_j \alpha_i \alpha_j k(\boldsymbol{x}_i,\boldsymbol{x}_j) \geq R^2. \tag{8.10}$$

SVDD can be run using a window-method, but is much more efficient if done globally to save computation (as with any algorithm). Of note, is that the kernel parameters must still be estimated, a detection threshold has to be chosen because R^2 vanished in the objective, and solving for the α_i coefficients is a quadratic programming problem. R^2 can be estimated from the training set by using the maximum from Equation 8.10, but ROC curves are most often generated by varying this threshold.

Minimizing L is a quadratic program for the α_i. Denote the Gram matrix for the kernels evaluated on the training set as K. Then the quadratic program is,

$$\min \; -Diag(K)^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T K\boldsymbol{\alpha}\text{: } \sum_i \alpha_i = 1, \alpha_i \geq 0. \tag{8.11}$$

So long as the rows of K are linearly independent, it is positive definite by definition of a Gram matrix. This is important, as this leads to global convergence for many quadratic programming solvers. For example, the Frank-Wolfe algorithm applied to the corresponding KKT conditions yields a global optimal, albeit with sub-linear or linear

convergence (if modified) [25]. Otherwise, K is only guaranteed semi-positive definite, and convergence for this formulation is only guaranteed to a local minimum.

This Banerjee, Burlina, and Diehl formulation [22] for SVDD assumes that a hyperplane exists such that the different classes can be separated without misclassification. That is, it is assumed that the resulting hypersphere truly encapsulates its training class and separates one class from another. Therefore, the training data must be separable from other classes and not contain outliers that would contaminate the optimization in that regard, and the training data must be from the known "background" class. In reality, these are very strong assumptions. In fact, SVDD for anomaly detection is a simplified form of the originating SVDD algorithm.

The original SVDD formulation from Tax and Duin [203] allowed for misclassification by using slack variables in the primal problem. Specifically, (8.1) was,

$$\begin{aligned} \min \quad & R^2 + \zeta \sum_{i}^{N} \xi_i \\ \text{subject to} \quad & \|\Phi(\boldsymbol{x}_i) - c\|^2 \leq R^2 + \xi_i, \quad i = 1, \ldots, N, \\ & \xi_i \geq 0, \qquad\qquad i = 1, \ldots, N, \end{aligned} \tag{8.12}$$

where ζ is a user-specified parameter that penalizes infeasibility relative to the hypersphere encapsulating the training data. Thus, a tighter hypersphere could be formed if the penalty was not too great relative to the amount of infeasibility. This formulation also gave the additional benefit of bounding the decision variables in the dual using ζ. That is, the corresponding Lagrangian dual to (8.7) is,

$$\begin{aligned} \max \inf \quad & \sum_{i}^{N} \alpha_i K_{ii} - \boldsymbol{\alpha}^T K \boldsymbol{\alpha} \\ \text{subject to} \quad & \sum_{i}^{N} \alpha_i = 1, \\ & 0 \leq \alpha_i \leq \zeta, \qquad i = 1, \ldots, N. \end{aligned} \tag{8.13}$$

Therefore, ζ has to be less than or equal to one, and any choice of penalty also serves to define the number of support vectors used such that a lower ζ enforces a tighter hypersphere

to the training data. ζ is often chosen to be $1/(n\eta)$, where η is an expected rejection rate in the training set and n is the number of exemplars in the training set [122]. Chang, Lee, and Lin [49] showed that the primal (8.1) is not a convex formulation due to the case of $R = 0$, but their resulting primal and dual formulations were generally equivalent assuming non-zero R and enforcing the convexity constraint on $\boldsymbol{\alpha}$.

8.1.2 Training Set and Spread Parameter Considerations.

Typically in SVDD, it is assumed that truth data is known and a training set is chosen from the set of background pixels [22, 199, 203]. With the class data known, it is easier to estimate an optimal σ using a Gaussian kernel. Polynomial kernels are not used as prevalently because their magnitude scales with both the magnitude of the data and the number of features in the data. Further, the Gaussian simplifies the formulation and is translation-invariant, and so the absolute positions of exemplars are not important [133]. In this research, the goal is to select the training set and any parameters in an unsupervised manner.

Banerjee, Burlina, and Diehl [22] proposed a cross-validation and minimax approach using false alarm rates to select a pseudo-optimal σ. In particular, given a number of training sets M of size n, the σ providing the least false alarm was approximated by that with the smallest average number of support vectors,

$$\hat{\sigma} = \min_{\sigma} \frac{1}{M} \sum_{i=1}^{M} \frac{SV_i}{n}, \tag{8.14}$$

where SV_i was the number of support vectors found for the i-th training set. This mean is an upper bound for the probability of false alarm. Wanga et al. [214] used an estimate based on the Fisher discriminant function to best separate the classes. Khazai et al. [121] scaled the maximum L_2 distance between all pairs of background samples. Later, they used,

$$\hat{\sigma}_i = \left(\sum_{i=1}^{p} Var\left(X^j\right) \right)^{1/2}, \tag{8.15}$$

where X^j denotes the j-th feature of the training class data X [122]. This was designed to model the standard deviation in a p-dimensional circular Gaussian probability density function with independent variables.

Gurram and Kwon [88, 89] proposed to find the optimal σ, and an optimal convex combination of kernels for SVDD by including σ and the kernel weights $\boldsymbol{\theta}$ in Equation 8.7. They derived gradients with respect to these parameters using the partial derivatives and used a reduced gradient method to find a descent direction. Again, however, it was assumed that class information was known. Further, without class information, such fine-tuning of the parameters could greatly overfit the hypersphere boundary.

Xiao, Liu, and Cao [219] built the training set by trying to find the boundary of the training class first. They used a function of k-furthest neighbors in the non-linear space to find those exemplars likeliest to be near the hypersphere boundary. They used a M-tree data structure to speed calculations, but also used the class information to pre-identify candidate training exemplars. Chu, Tsang, and Kwok [57] used core sets to iteratively build a training set. Specifically, they randomly sampled a core set from the training class and estimated a center. Then, exemplars from the core set within some radius were added into a training set, and the center was re-estimated. This was done iteratively until the size of the training set was some desired cardinality. As this generation does not emphasize the boundary, and because class information is assumed to be known, it is not useful for this research.

Hua and Ding [101] proposed to incrementally build the training set, so that very large data sets could be used without limiting training to a small sample of the data. To do this, they noted that only background exemplars that violate the Karush-Kuhn-Tucker (KKT) conditions at any step need to be incorporated. That is, an initial training set could be used, and then any remaining training exemplars would either be within the hypersphere boundary, on it, or outside of it. Those outside of it would then be incrementally added to expand the boundary. Tavakkoli et al. [202] developed a similar techniques for use on

video frames, where single exemplars were added to the training set. These incremental approaches are only useful if there is high confidence in class labels.

8.1.3 SemiBoost.

In this research, class information is assumed to be unknown. However, it may be that certain exemplars can be assumed to be anomalous or background with high confidence. In this case, the data could be split into three subsets: likely anomalous, likely background, and unknown. The unknown exemplars can remain unlabeled, while the others can be labeled with their likely class. Such an approach allows for a semi-supervised training method.

Mallapragada et al. [152], inspired by the power of boosting methods in a fully supervised case, developed the SemiBoost algorithm to boost classification on a mix of labeled and unlabeled data. Let y_i^u denote the class label prediction for the i-th exemplar that is unlabeled and y_i^l denote the class label for the i-th labeled exemplar. Here, class labels are either -1 or 1. Then, a reasonable objective is for similar points to be labeled the same. They developed the objective,

$$\sum_{i,j=1}^{n_u} S_{i,j} \exp\left(y_i^u - y_j^u\right) + C \sum_{i=1}^{n_l} \sum_{j=1}^{n_u} S_{i,j} \exp\left(-2y_i^l y_j^u\right), \tag{8.16}$$

for this purpose, where $C = n_l/n_u$ and $S_{i,j}$ is the similarity between x_i ad x_j. Now, after assuming T weak classifiers $h_t(x)$ are used to build a stronger classifier $H(x) = \sum_{i-1}^{T} \alpha_t h_t(x)$, they showed that optimizing Equation 8.16 is the same as minimizing,

$$\sum_{i=1}^{n_u} \exp\left(-2\alpha h_i\right) p_i + \exp\left(2\alpha h_i\right) q_i, \tag{8.17}$$

where

$$p_i = \sum_{i=1}^{n_l} S_{i,j} e^{-2H_i} \delta(y_j, 1) + \frac{C}{2} \sum_{j=1}^{n_u} S_{i,j} e^{H_j - H_i}, \tag{8.18}$$

and

$$q_i = \sum_{i=1}^{n_l} S_{i,j} e^{2H_i} \delta(y_j, -1) + \frac{C}{2} \sum_{j=1}^{n_u} S_{i,j} e^{H_i - H_j}. \tag{8.19}$$

Additionally, they derived the optimal weights for the weak classifiers as,

$$\alpha = \frac{1}{4} \ln \frac{\sum_{i=1}^{n_u} p_i \delta(h_i, 1) + \sum_{i=1}^{n_u} q_i \delta(h_i, -1)}{\sum_{i=1}^{n_u} p_i \delta(h_i, -1) + \sum_{i=1}^{n_u} q_i \delta(h_i, 1)}, \tag{8.20}$$

where δ is an indicator function. They proved that using these weights the objective function follows an exponential decay [152]. In order to best reduce the objective function each iteration, the weights $|p_i - q_i|$ are used to select the most confident unlabeled data exemplars, and the algorithm stops if $\alpha_t \leq 0$ due to such a low performance classifier breaking the convergence properties. The SemiBoost algorithm is shown as Algorithm 8.1.

Algorithm 8.1 SemiBoost [152]

1: Let X be a set of $(n_l + n_u)$ exemplars, where n_l are labeled and n_u are unlabeled.
2: Compute the pairwise similarity matrix between all exemplars, S_{ij}.
3: $0 \leftarrow H(X)$.
4: **for** $t = 1 : T$ **do**
5: For each exemplar i in X, compute p_i and q_i using Equations 8.18 and 8.19.
6: $sign(p_i - q_i) \leftarrow z_i$.
7: Sample exemplar x_i for use in training the weak classifier according to weight $|p_i - q_i|$.
8: Train weak classifier $h_t(x)$ using sampled examples and class labels z_i.
9: Compute α_t using Equation 8.20.
10: $H(X) + \alpha_t h_t(X) \leftarrow H(X)$
11: **end for**

8.2 Unsupervised Training Set Generation and Parameter Optimization

SVDD is typically not performed without known class information, due to its fitting of a decision boundary to the supports of the training set. The algorithm itself is fairly efficient with good results, and estimates a geometry for the data. Therefore, it may

be beneficial to develop SVDD for the unsupervised case. In order to do this, one consideration is that finding the supports for the background class may be primarily a function of finding those exemplars with a large margin to the anomalous class. k-furthest neighbor and incremental methods are likely too inefficient without class information to guide the approach. Unsupervised support vector machines often try to find a largest margin, but this is problematic if an image is more cluttered. In other words, an image with water, land, and ships likely has the largest margin between the water pixels, and the land and ship pixels. The land class acts like a weak anomaly class, whereas the ship pixels are the strong anomaly class of interest.

Instead, an entirely different approach can be considered. The BACON algorithm (Section 3.11.5) screens outliers as a function of the Mahalanobis distance, making it very efficient. However, even with variation of its parameters it does not always perfectly separate background and anomaly classes. Further, results can be inconsistent across image types if using a single, fixed set of parameters. Consider the results for the percentage of pixels detected at outliers for a fixed $v = 30$ degrees of freedom, shown in Figure 8.1, varied by significance. ARES1C and ARES2C have no targets, yet BACON is prone to false positives on those images.

Despite these findings, BACON is useful in finding background. That is, a screening most often detects targets and the more anomalous background pixels. This can be used to an advantage by double screening. Figure 8.2 depicts the first part of this concept. First, a normal iteration of BACON is performed to identify potential anomalies. After experimentation across several images, it seemed best to do this at the $\alpha = 0.05$ significance with 20 degrees of freedom. This iteration separates clean background from the rest of the pixels. Next, BACON is applied to the pixels identified as anomalous in the first iteration, but at a changed sensitivity. This serves to separate the more anomalous background from the real anomaly class. In practice, for best results across images, $\alpha = 0.1$ and 10 degrees

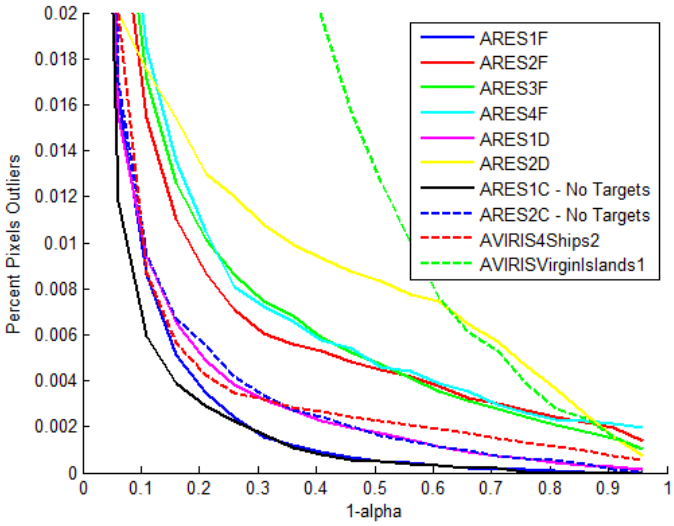


Figure 8.1: BACON with $v = 30$.

of freedom were used for the cut-off in the second iteration. By itself this could be used as an anomaly detection algorithm, but in practice this does not always identify all anomalous pixels, nor does it always remove all false positives from the estimate. Instead, it is used here to begin to characterize the background and anomalous classes.

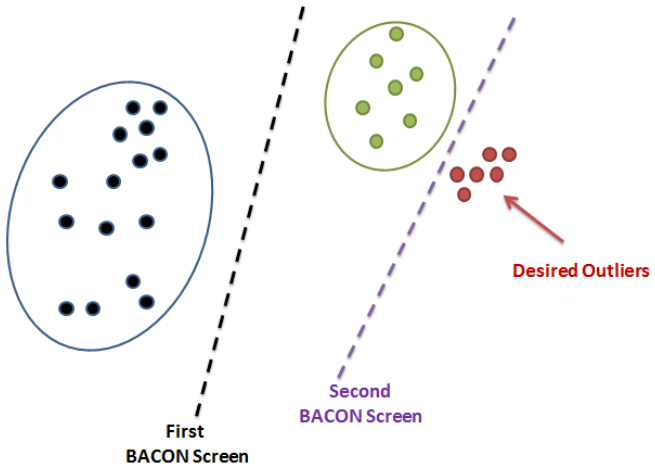
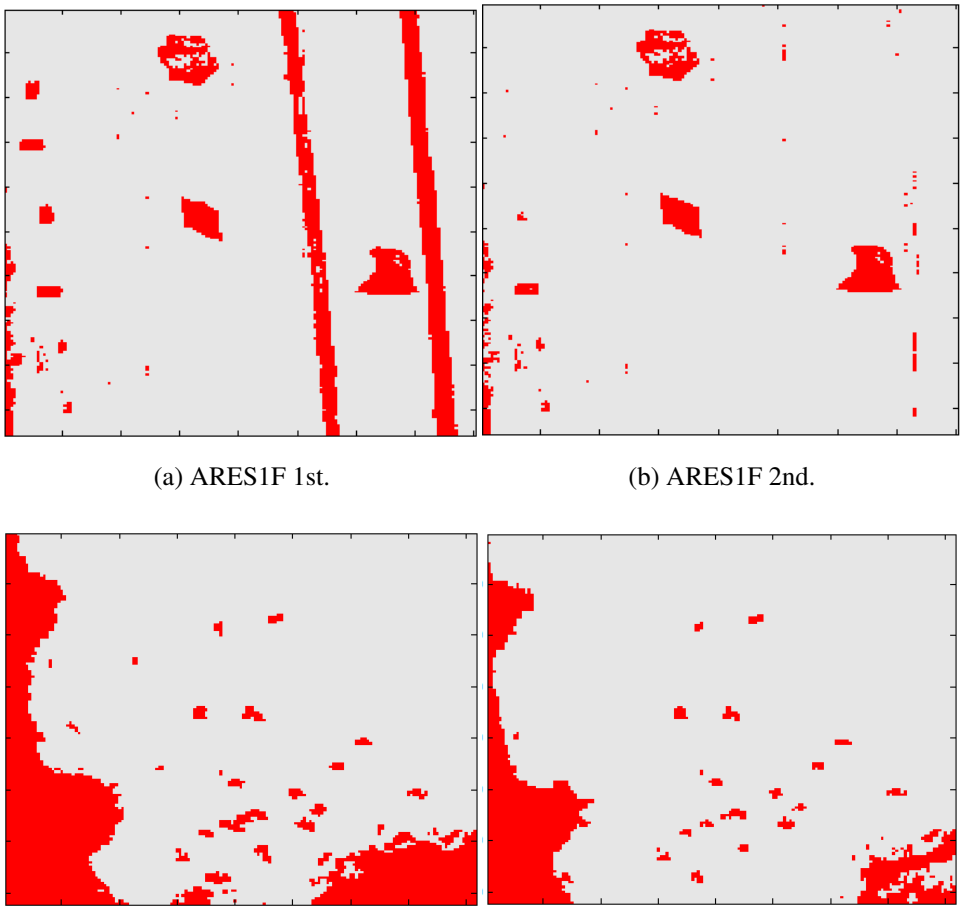


Figure 8.2: BACON Double-Screening Approach.

Figure 8.3 shows results at the end of each iteration for ARES1F and VirginIslands1. Whereas the results are very desirable for ARES1F, the nature of the land and water pixels in the VirginIslands1 image makes it more difficult to screen out false positives, resulting in a very large number. In such a case, a skeleton technique that could help to mitigate this is necessary. Additionally, if very few pixels are identified as anomalous in the second iteration, then the image can be considered to have no anomalies. This resolves one of the main issues with any form of SVDD, in that by nature of the training set's hypersphere boundary, a target class is always assumed.

In Section 7.3.1, a large-scale affinity propagation technique based on landmark points was presented, denoted as LAP. It was shown that this method was adept as finding the boundaries of classes in the data, and that a related partitioning counterpart, PLAP, was adept at finding a centroid skeleton for these boundaries. Here, it is proposed to use these algorithms to form skeletons on the results of the previous BACON screening, in order to yield probable boundary solutions with which to train an unsupervised SVDD. Taking into consideration the nature of the hypersphere boundary, it is more desirable to use LAP on those pixels not identified as anomalous, so as to provide a larger background support. Even in the supervised case, LAP could provide a better boundary than the standard method of randomly sampling from the background.

Any boundary is still dependent on the choice of kernel. In order to better select the kernel, but again without class information, the technique from Section 7.1.3 can be used. However, this requires some labeled subset with which to calculate the kernel fisher criterion. To do this, LAP and PLAP can again be used. Similar to before, LAP performed on those pixels not identified as anomalous by the BACON screening yields exemplars that shape the boundaries of the background. Meanwhile, PLAP performed on the potential anomalies yields exemplars that represent the shape of the anomaly class, but that are not necessarily nearest the boundaries. This helps to limit the amount of over-training for the



(a) ARES1F 1st. (b) ARES1F 2nd.

(c) VirginIslands 1st. (d) VirginIslands 2nd.

Figure 8.3: BACON Double Screening Results.

kernel. This is depicted in Figure 8.4. Given these considerations and frameworks, a full unsupervised SVDD (USVDD) method is presented next.

8.3 Unsupervised SVDD (USVDD)

The unsupervised SVDD algorithm is given as Algorithm 8.2. After each BACON iteration, if there are no potential anomalies or if the number is below some threshold as a function of the percentage q of the image pixels, then it is determined that the image has

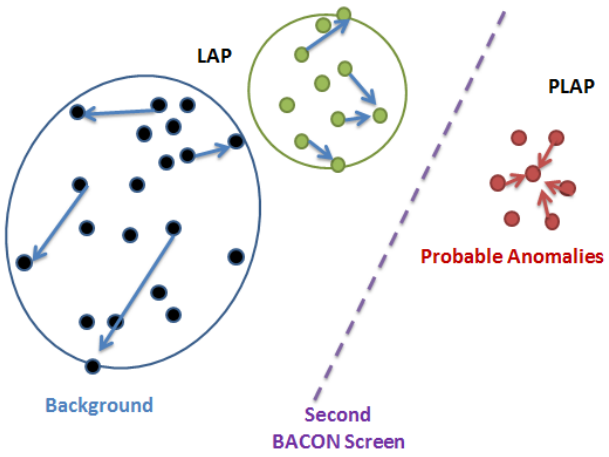


Figure 8.4: Landmark Generation for Optimal Kernel.

no anomalies. As SVDD always fits to a training set, this is a necessary step. On ARES1C, only 124 pixels, or 0.0057% of the image's pixels are detected as potentially anomalous on the first BACON iteration. On ARES2C, another image with no targets, 0.0682% are falsely detected after the second screening. Other ARES images generally have 0.01 to 0.03% detected. Therefore, unfortunately, this threshold is not entirely straightforward, but $q = 0.01$ may work well in most cases.

To select an optimal kernel, ten Gaussian candidates were generated for the optimization according to log scale, using 0.001 and one half of the squared range between minimums and maximums in the dataset as the endpoints. This set, and the corresponding optimal linear combination, is shown in Figure 8.5 for ARES2D. Larger numbers of candidates, and including a first, second, and third-order candidate polynomial kernel were tested as well, but most often only the Gaussian kernels were chosen to have non-zero contribution in the optimization. Using only Gaussian candidates can also be advantageous in that the optimal kernel is itself a Gaussian (treating the candidates as independent).

Algorithm 8.2 Unsupervised SVDD

Let X be the set of N pixels in the image, and $0 < q << 1$.

Perform BACON with $\alpha = 0.05$ and $v = 20$ d.o.f. on X, yielding a subset X_b of probable background pixels and a subset X_a of potential anomalies.

if $X_a = \emptyset$ or $|X_a| < qN$ **then**

    Declare no anomalies, and end.

end if

Perform BACON with $\alpha = 0.1$ and $v = 10$ d.o.f. on X_a, yielding a subset X_a^* of potentially anomalous pixels.

if $X_a^* = \emptyset$ or $|X_a^*| < qN$ **then**

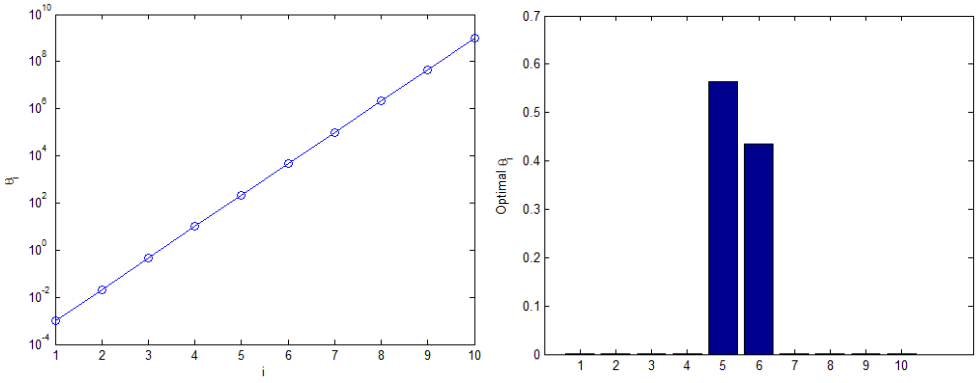
    Declare no anomalies, and end.

end if

$X_b \leftarrow LAP(X_b \cup (X_a \setminus X_a^*))$.

(Optional: Optimal Kernel) $X_a^* \leftarrow PLAP(X_a^*)$. Labeling X_b and X_a^* as distinct classes, solve for the optimal kernel.

$X_a \leftarrow SVDD(X_b)$.



(a) Candidates. (b) Optimal Combination.

Figure 8.5: Kernel Selection: ARES2D.

To generate comparisons for USVDD, three supervised SVDD methods were applied. The first two were a baseline SVDD with settings as employed by Taitano, Gaier, and Bauer [199], and framework developed by Banerjee, Burlina, and Diehl [22]. 500 training exemplars were randomly chosen from known background and the Lagrangian multipliers were bounded by $1/(0.01N)$, where again, 0.01 represented an expected false alarm rate. In the first case, a single training set was used where σ was set as a function of the average distance to the mean of X_b. In the second, the minimax approach was used to find a best σ over a candidate set of 40 values on a range of 20 to 8000, where $M = 3$. As can be seen in Table 8.1, even with these small numbers, this approach greatly increased the expense of the algorithm. These two standard SVDD approaches were replicated 10 times due to their inherent randomness, and a mean was taken.

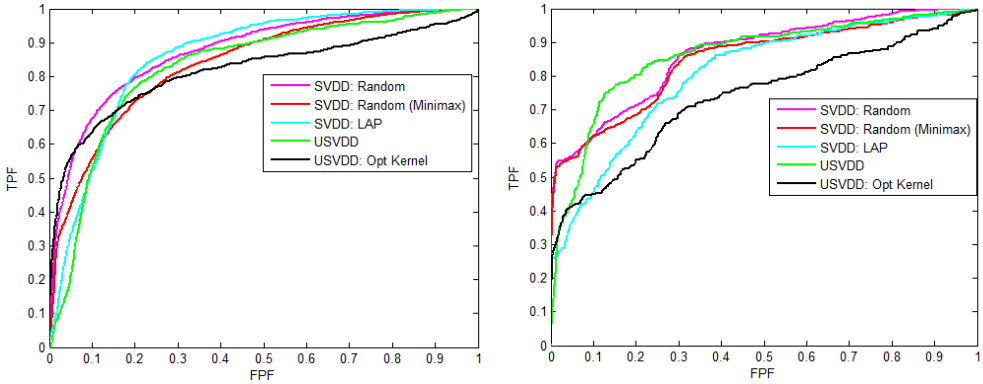
LAP was also used to generate the training data for supervised SVDD in a third variation, where the clustering took place on known background using the truth data. Again, σ was set as a function of the average distance to the mean of X_b. The USVDD algorithm with and without optimal kernel, and using $m = 1000$ in the LAP algorithm

was also run. The computational expense of the algorithms, in seconds, on seven test problems are shown in Table 8.1, where experiments were done using a Intel™ Core i7 CPU Q840@1.87 GHz, 64-bit OS, with 8 GB RAM. The LAP algorithm and use of multiple training sets are the primary contributor to computational expense. Although larger skeletons may be desirable for LAP, its expense scales with m. Smaller m yields a smaller decision boundary and so is not necessarily desirable.

	Supervised - Random	Supervised - Random (Minimax)	Supervised - LAP	USVDD	USVDD w/Optimal Kernel
ARES1F	37.52	1187.44	86.55	83.06	99.04
ARES2F	46.00	1332.98	157.28	125.12	225.65
ARES3F	35.26	1463.45	130.73	103.05	120.96
ARES4F	35.20	1393.04	56.84	61.68	81.98
ARES1D	44.09	1112.29	240.63	171.40	177.66
ARES2D	35.83	903.82	96.63	75.09	79.93
AVIRISVirginIslands1	32.15	1052.50	82.66	70.47	145.79

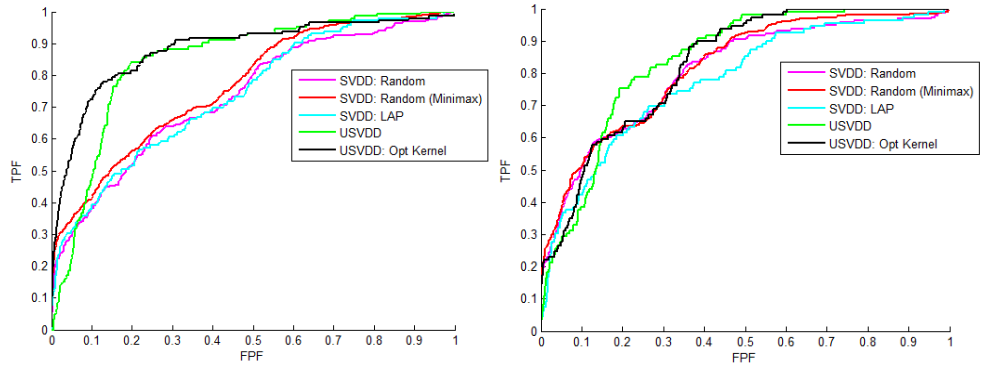
Table 8.1: SVDD Computational Time Comparisons.

ROC curves for these four techniques are shown in Figures 8.6 and 8.7. It is clear that the optimal kernel often over fits, yet it performs the best on ARES3F. USVDD is highly competitive with all of the algorithms, often out-performing its supervised counterparts, although further decreases in FPF rates could prove very beneficial. It also outperforms its initial BACON estimate, as evidenced by the VirginIslands1 image (ref: Figure 8.3). In this case, the land class is also anomalous when compared to the water. Therefore, the double screening helps to separate the true targets from the entirety of the background. Using Equation 7.18 to set σ instead of using a minimax approach appears to be very competitive and saves a great deal of computational expense. The values for σ were not typically equal between the two methods, which suggests multiple competitive candidates. This phenomena could also be partly due to the random nature of the baseline SVDD method.



(a) ARES1F.

(b) ARES2F.

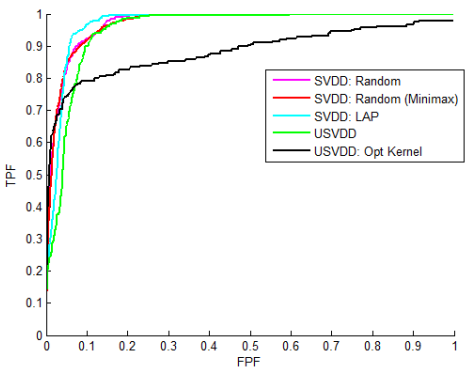


(c) ARES3F.

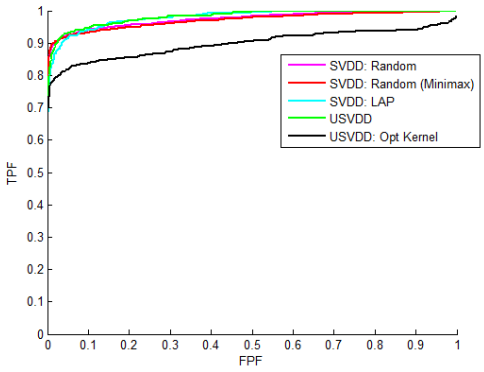
(d) ARES4F.

Figure 8.6: SVDD Comparison: Forest Scenes.

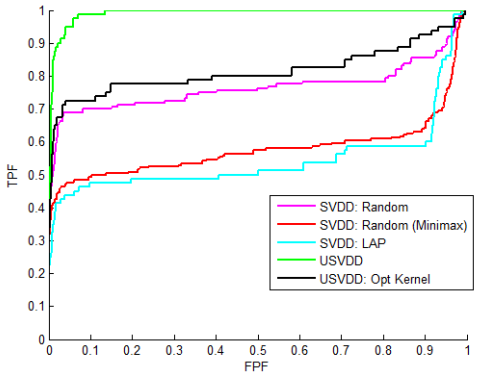
Next, a SemiBoost approach to USVDD was taken. The LAP and PLAP approach from the kernel optimization was used to yield a set of labeled pixels. To form a third class of unlabeled pixels, those pixels not in the labeled classes were clustered again using LAP. Given these three sets of pixels, SemiBoost could be performed directly. Unfortunately, the sparsity of the affinity propagation skeletons are a detriment in SemiBoost, as there are very few labeled and unlabeled exemplars under consideration relative to the total number of pixels. This provides weak classifiers that do not contribute a large amount of useful information to the overall strong classifier. Results for the SemiBoost algorithm using



(a) ARES1D.



(b) ARES2D.



(c) VirginIslands1.

Figure 8.7: SVDD Comparison: Desert and Water.

SVDD for the weak classifier are shown in Figure 8.8 for ARES1F. $m = 2000$ in the LAP sub-algorithm for the background estimates was also used, but this did not improve the ROC curves. Different sampling methodologies were also employed to try and affect the classifiers, to include sampling background and probable background pixels only, but this had little effect on the ROC curve. Again, this suggests a different, fuller skeleton approach is required for the SemiBoost technique to work here. Increasing the iterations does improve the ROC curve, but the algorithm with five iterations took 192 seconds while

for 20 iterations it took 256 seconds. Relative to USVDD without a boosting approach, the computational expense is not a worthwhile trade-off.

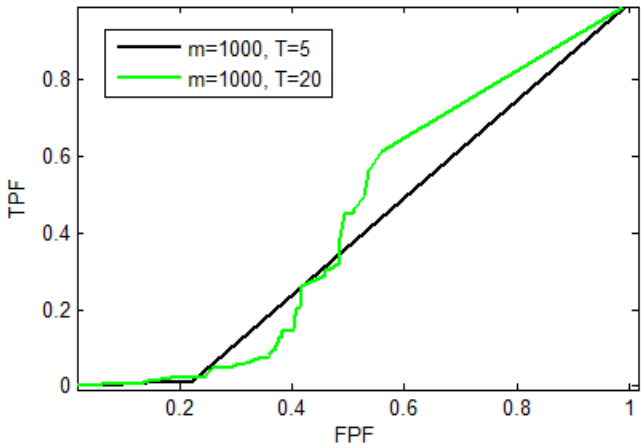


Figure 8.8: SemiBoost USVDD.

As a final investigation for a semi-supervised SVDD, the predictions from IGFAAD (Table 6.16) were input to USVDD, using predicted background as assumed background and predicted target as assumed target. Then LAP was used on the assumed background to generate the training set for USVDD. Due to the good performance of IGFAAD, this was intended as an approximate semi-supervised technique or fusion to try and boost the IGFAAD algorithm's performance. With and without the optimal kernel step, resulting ROCs generally had solutions less desirable than the original IGFAAD predictions, where TPFs were as high only if allowing for increased FPFs. One of the better performing images was VirginIslands1, where this ROC curve is shown in Figure 8.9. Even still, in the original IGFAAD results, the FPF was 0.07 and the TPF was 0.95. ARES3F is also shown, and it is clear that this fusion is not desirable, as the original results had a TPF of 0.8552 and FPF of 0.07. This could be because the background pixels do not all truly fall within an exclusionary hypersphere in the non-linear space. This further shows that

although USVDD is competitive with its supervised SVDD counterparts, the GFAAD and IGFAAD methods from Chapter 6 operate at a much better point in the TPF and FPF space.

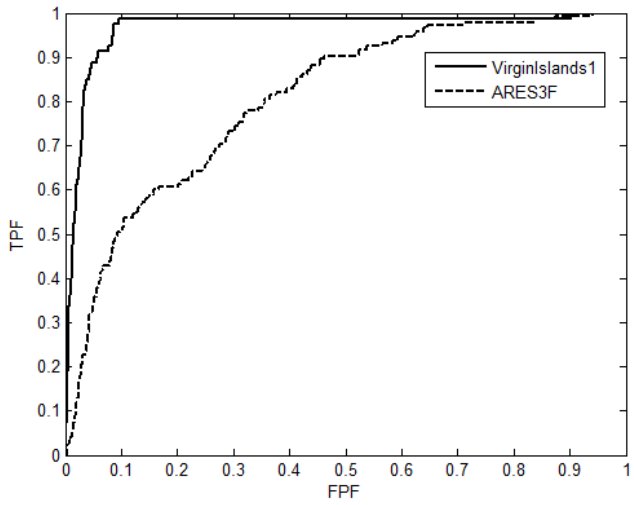
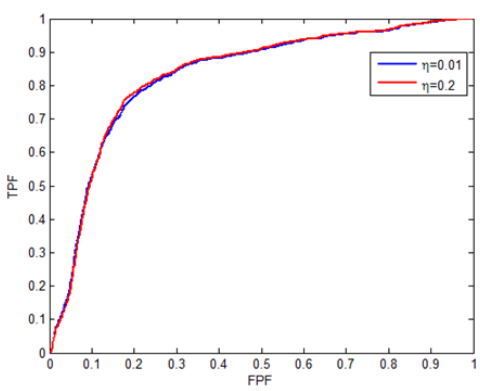
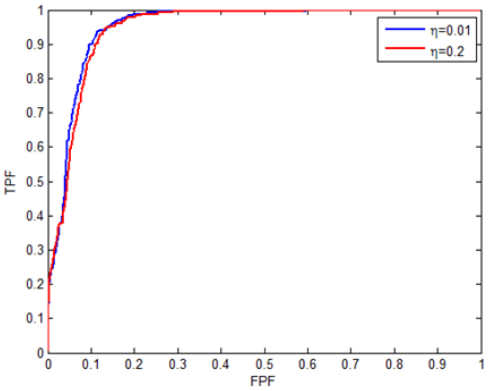


Figure 8.9: Fused IFGAAD and USVDD Results.

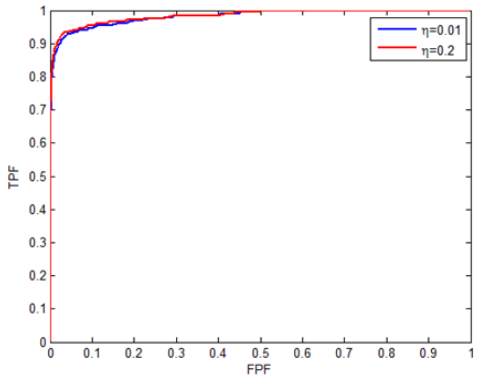
Investigating the sensitivity of USVDD to the bound ζ on the dual variables is also necessary. Recall, this is often chosen as $1/(\eta N)$, where η is an expected false alarm rate. 0.01 was used previously based on the bound used by Taitano, Gaier, and Bauer [199]. The results of USVDD are fairly invariable to this bound. In fact, varying η from 0.1 to 0.2, many results are identical. Figure 8.10 shows the three images where ROC curves differ slightly, depicting the range of η, and clearly there is only small variation. This may be partly due to the fact that the LAP background estimate used to train is so small.



(a) ARES1F.



(b) ARES1D.



(c) ARES2D.

Figure 8.10: USVDD Dual Variable Bound Comparison.

IX. Summary of Contributions

9.1 Review

In this research, approaches to better understanding multivariate data and their anomalies were developed. Specifically, the following contributions were made:

1. An unsupervised algorithm was developed to find noisy features, using factor analysis.

2. HSI truth masks were analyzed so as to better measure true and false positives.

3. Improvements were made to an existing n-dimensional visualization, where it was re-formulated as an optimization problem and several techniques were developed to provide solutions.

4. A new global anomaly detection algorithm was developed utilizing a fusion of spectral, spatial, and SNR information, and factor analysis, filtering, and zero-bin histogram techniques.

5. A non-linear form of the new anomaly detection algorithm was developed, where a data skeleton methodology was tested and incorporated to make it more computationally competitive with its linear counterpart.

6. An unsupervised support vector anomaly detection algorithm was developed, utilizing clustering and screening methods that were also newly developed.

In Chapter 4, the factor-analysis based feature selection technique was developed to identify noisy features and absorption bands successfully. Specific variance thresholding was shown to remove undesirable bands from HSI. This technique is very efficient, and proved relatively robust to the number of features under consideration and to different types

of data. In particular, it was shown to also work on a dataset with 3,000 noise features, where all 3,000 were clearly identified. The truth masks for many HSI images were also evaluated in depth, revealing that many pixels typically treated as background or target to fit an algorithmic need, are in fact more like background.

Development of a new n-dimensional visualization formulation and method was provided in Chapter 5, where this visualization is highly intuitive. Objective functions for the formulation were provided for both supervised and unsupervised data. Heuristics with which to optimize the visualization in order to reveal class and object structure were presented and demonstrated across a set of natural and HSI data sets.

Chapter 6 developed and discussed the GFAAD framework. The use of factor analysis in a global detection algorithm proved to be highly competitive, and more robust across image types, than existing algorithms. The resulting framework used many concepts from AutoGAD and MPCA in a revised manner, creating a process that adapts to individual score maps under consideration. This adaptation was enabled by determination of three categories of factor score maps. Spectral, spatial, and SNR information was fused within the algorithm to dynamically adjust to specific factors of an image. The new algorithm removed randomness from the detection, increased full-pixel anomaly determination, and provided interpretable components. Additionally, a version of the algorithm with only slightly increased false positives was shown to be more computationally efficient than existing methods. A single set of parameter settings was established for desirable performance across a suite of 19 HSI images with varying characteristics and complexities.

Chapter 7 introduced a new kernel factor analysis technique, and presented comparisons of various skeleton generation techniques in order to make it computationally competitive. It was shown that these non-linear methods can provide useful mappings with which to identify anomalous pixels, and that sacrificing skeleton size to save computational expense does not significantly affect results, to a point. In fact, these reduced skeletons were

shown to sometimes be more useful, in that there are fewer mappings to evaluate and, at times, fewer false positives. The IGFAAD framework was expanded to the kernel case, and shown to outperform and be comparatively efficient to other non-linear techniques.

Unsupervised techniques for the SVDD algorithm were developed in Chapter 8. Although not as powerful as the factor analysis methods presented previously, they were shown to perform competitively and in some cases better than standard supervised SVDD algorithm variants. Additionally, these unsupervised techniques removed randomness from the SVDD framework and were comparatively efficient. A unique training set generation technique was developed using a new hybrid of BACON and AP clustering methods.

9.2 Insights

Factor analysis, a method not prevalently used in anomaly detection, proved to be a powerful technique. Varimax rotated factors yielded certain clean mappings against which anomalies were more easily identifiable. The fusion of spectral, spatial, and SNR information, as in methods such as AutoGAD and MPCA, also proved to be highly useful. Using this set of characteristics allows an algorithm to find locations in the spectrum where certain pixels are significantly different from the majority, to evaluate whether a pixel is also distinct from its neighbors, and to determine which mappings may be useful for the detection problem.

Factor analysis was also very useful in determining noisy features. The method developed is not entirely robust to sparse outliers, however, as evidenced by the retention of certain spectral bands in HSI where the sensor generated a line of erroneous pixels. These are easily visually identified, but are not clear from the specific variance threshold.

Many of the methods developed in this research, such as IGFAAD and USVDD, can likely be further optimized. Their current high performance is promising towards this end, and the factor analysis-based methods specifically were shown to be better performing than many existing algorithms. Although the non-linear methods also exhibit good performance,

their current computational expense in comparison to well-performing linear methods may still prohibit their use under certain conditions. The new global affinity clustering techniques developed here are simple, yet very accurately find the shape and boundaries of data. Most of the techniques developed have the additional advantage that they are not random, and are entirely unsupervised.

Using hyper-radials to visualize data was seen to be extremely effective. Although the formulation to find an optimal set of axes is complex, pseudo-optimal solutions often reveal information about class structure or outliers in the data. These more advanced methods easily extend back their original purpose of evaluating design solutions in multi-objective optimization as well.

It is also important to note here that image size does not appear to be as important to the success of the algorithms developed here as one might imagine. Variation in how images ARES1D and the AVIRIS imagery perform makes it clear that characteristics such as PA SNR of mappings are more heavily influenced by the make-up of the scene in the image.

9.3 Potential Future Research

9.3.1 HSI Band Selection Refinement.

The factor analysis-based band selection algorithm from Chapter 4 did not always detect bands containing sensor error, where a partial line of pixels was corrupted. Future research into automating removal of such bands could prove useful, as it was shown that they contributed to the FPF of the GFAAD framework.

9.3.2 GFAAD Refinement.

The GFAAD and IGFAAD algorithms in Chapter 6 showed great promise across a variety of images and sensors in identifying full-pixel anomalies and maintaining relatively low false positives. However, the SNR and smoothing parameters used in the frameworks have significant interactions and can influence the factor maps for images with different

characteristics in significant, various ways. A Robust Parameter Design (RPD) analysis is warranted, as is a more thorough investigation into characteristics other than SNR with which to adapt the algorithm to different image or factor types. Images with very high score ranges and high SNR would benefit from increased smoothing, for example, but the adjusted settings need to be automated such that those images with mostly low SNR and score factors maps are not affected. The RPD could serve to reduce the number of parameters while minimizing loss in performance, as well as to boost performance across different image types. This is also true for the kernel version from Chapter 7, KIGFAAD. Additionally, more research into dynamic processes that further adapt to complexities of the image, such as soft anomaly classes or not well-separated anomalies, should be performed.

9.3.3 Finding Better Unsupervised Boundaries for SVDD.

The BACON screening and affinity propagation methods provided very reasonable training sets for an unsupervised SVDD algorithm. However, the affinity propagation in particular adds a significant amount of computation. Ways to speed this skeleton generation could be very useful, for this purpose and for skeleton generation in non-linear applications in general. The optimal kernel technique developed here typically over-trained, but did not always. Development of a slack term in the formulation could be useful to ensure that the kernel boundary does not over-fit to estimated class labels. Ways to decrease false positives in general should also be investigated for the unsupervised algorithm.

9.3.4 Improving Non-Linear Anomaly Detection.

Despite the relative efficiency of using skeletons vice trying to estimate the exact, full kernel eigenvectors, non-linear components are still somewhat inefficient when compared to their linear counterparts. Investigation into a smarter fusion of non-linear components, or of generation of a skeleton more robust to differing image characteristics and outliers is warranted. The current skeleton generation can prove to be very sensitive. Better mappings are necessary in order for there to be sufficient performance improvement to warrant the

additional computation time. The optimal kernel problem is still very much open, and could a key to this goal as well.

Investigation into whether USVDD can even compete with linear methods is also warranted. That is, there is the possibility that some of the HSI images do not have background and target classes that truly separate by a hypersphere boundary in a higher-dimensional space. If it can be proven somehow that they can be separated in such a manner, then further investigation on how to improve the SVDD framework should be performed.

9.4 Conclusion

Simpler methods, such as linear factor analysis, have been shown to be very powerful across different types, sizes, and complexities of data. When fused with spatial information by filtering, and SNR information by using zero-bin histograms, the linear methods can be made to act very much like non-linear methods. Similar fusion of spatial and SNR information was also shown to be powerful for non-linear methods.

Several new approaches to anomaly detection, visualization, clustering, and feature selection were developed in this research. Many are new combinations of, or frameworks for, existing methodologies. It has been shown that these new approaches are generally efficient, and highly competitive with state-of-the-art algorithms in their respective areas. Further, they have been shown to perform well across a variety of HSI data. Experimentation has indicated that these approaches may not yet be at their optimal operating point, yielding even greater promise for future development.

Bibliography

[1] "Hyperspectral Remote Sensing Scenes". Website, May 2013. URL http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

[2] "AVIRIS Spectra". Website, February 2014. URL http://aviris.jpl.nasa.gov/aviris/spectrum.html.

[3] "Data for Matlab hackers". Website, February 2014. URL http://www.cs.nyu.edu/~roweis/data.html.

[4] "The MNIST Database". Website, February 2014. URL http://yann.lecun.com/exdb/mnist/.

[5] "Reflective Optics System Imaging Spectrometer (ROSIS)". Website, March 2014. URL http://www.opairs.aero/rosis_en.html.

[6] "SpecTIR Free Data Samples". Website, February 2014. URL http://www.spectir.com/free-data-samples/.

[7] A. Hyvrinen, J. Karhunen and E. Oja. *Independent Component Analysis*. John Wiley & Sons, third edition, 2001.

[8] Achlioptas, D. "Database-friendly random projections". *Proc. of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems*, 274–281. 2001.

[9] Achlioptas, D. "Database-friendly random projections: Johnson-Lindenstrauss with binary coins". *Journal of Computer and System Sciences*, 66:671–687, 2003.

[10] Achlioptas, D. and B. Chazelle. "Approximate nearest neighbor and the fast Johnson-Lindenstrauss transform". *Proc. of the thirty-eighth annual ACM symposium on theory of computing*, 557–563. 2006.

[11] Agrawal, G., K.E. Lewis, and C.L. Bloebaum. "Intuitive Visualization of Hyperspace Pareto Frontier". *44th AIAA Aerospace Sciences Meeting and Exhibit*. Reno, Nevada, January 2006.

[12] Ahmed, Y. *Multiple random projection for fast, approximate nearest neighbor search in high dimensions*. Master's thesis, University of Toronto, 2004.

[13] Alpern, B. and L. Carter. "Hyperbox". *Proc. IEEE Conference on Visualization (VIS '91)*, 133–139. San Diego, CA, 1991.

[14] Altmann, Y., N. Dobigeon, and J. Tourneret. "Nonlinearity detection in hyperspectral images using a polynomial post-nonlinear mixing model". *IEEE Transactions on Image Processing*, 22(4):1267–1276, Apr 2013.

[15] Arif, M. and S. Basalamah. "Similarity-Dissimilarity Plot for High Dimensional Data of Different Attribute Types in Biomedical Datasets". *International Journal of Innovative Computing, Information and Control*, 8(2):1275–1297, 2012.

[16] Arya, S., D. Mount, N. Netanyahu, R. Silverman, and A. Wu. "An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions". *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, 573–582. Arlington, VA, Jan 1994.

[17] Ashton, E.A. and A. Schaum. "Algorithms for the detection of sub-pixel targets in multispectral imagery". *Photogram. Eng. Remote Sens.*, 723–731, Jul 1998.

[18] Bach, F. and M. Jordan. "Kernel Independent Component Analysis". *Journal of Machine Learning Research*, 3:1–48, Jul 2002.

[19] Bache, K. and M. Lichman. "UCI Machine Learning Repository". Website, 2013. URL http://archive.ics.uci.edu/ml.

[20] Baghbidi, M., K. Jamshidi, A. Nilchi, and S. Homayouni. "Improvement of anomaly detection algorithms in hyperspectral images using discrere wavelet transform". *Signal and Image Processing*, 2(4):13–25, 2011.

[21] Baghbidi, M., K. Jamshidi, A. Nilchi, and S. Homayouni. "Improvement of anomaly detection algorithms in hyperspectral images using discrete wavelet transform". *Signal and Image Processing*, 2(4):13–25, Dec 2011.

[22] Banerjee, A., P. Burlina, and C. Diehl. "A support vector method for anomaly detection in hyperspectral imagery". *IEEE Transactions on Geoscience and Remote Sensing*, 44(8):2282–2291, Aug 2006.

[23] Basener, B., E. Ientilucci, and E. Messinger. "Anomaly detection using topology". *SPIE Defense and Security Symposium*. SPIE, 2007.

[24] Basener, W. and D. Messinger. "Enhanced detection and visualization of anomalies in spectral imagery". *SPIE Defense, Security, and Sensing*. SPIE, 2009.

[25] Bazaraa, M., H. Sherali, and C.M. Shetty. *Nonlinear Programming*. John Wiley & Sons, third edition, 2006.

[26] Bengio, Y., P. Vincent, and J. Paiement. *Learning Eigenfunctions of Similarity: Linking Spectral Clustering and Kernel PCA*. Technical Report Technical Report 1232, Department of Information Technology and Operational Research, University of Montreal, February 2003.

[27] Bertini, E., A. Tatu, and D. Keim. "Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization". *IEEE Transaction on Visualization and Computer Graphics*, 17(12):2203–2213, December 2011.

[28] Bguin, C. and B. Hulliger. "The BACON-EEM algorithm for multivariate outlier detection in incomplete survey data". *Survey Methodology*, 34(1):91–103, 2008.

[29] Bhattacharyya, A. "On a measure of divergence between two statistical populations defined by probability distributions". *Bulletin of the Calcutta Mathematical Society*, 35:109–109, 1943.

[30] Bihl, T., M. Friend, F. Mindrup, and K. Bauer. *Unsupervised Image Quality Estimation*. Technical report, Air Force Institute of Technology, 2011.

[31] Billor, N., A. Hadi, and P. Velleman. "BACON: blocked adaptive computationally efficient outlier nominators". *Computational Statistics and Data Analysis*, 34:279–298, 2000.

[32] Blum, A. *Subspace, Latent Structure and Feature Selection*, chapter Random projection, margins, kernels, and feature-selection, 52–68. Springer Berlin Heidelberg, 2006.

[33] Borengasser, M., W. Hungate, and R. Watkins. *Hyperspectral Remote Sensing: Principles and Applications*. CRC Press, Boca Raton, FL, 2010.

[34] Borghys, D., E. Truyen, M. Shimoni, and C. Perneel. "Anomaly Detection in Hyperspectral Images of Complex Scenes". *Proceedings of the 29th EARSel Symposium*. Chania, Greece, 2009.

[35] Boros, E. and P. Hammer. "Survey-Boolean Optimization". *Discrete Applied Mathematics*, 123(1):155–225, 2002.

[36] Bottegal, G. and G. Picci. "A note on Generalized Factor Analysis models". *50th IEEE Conference on Decision and Control and European Control Conference*, 1485–1490. Orlando, FL, Dec 2011.

[37] Bradley, P. and U. Fayyad. *Refining Initial Points for k-Means Clustering*. Technical Report MSR-TR-98-36, Microsoft Research, Microsoft Corporation, 1998.

[38] Brandes, U. and C. Pich. "Eigensolver Methods for ProProgress Multidimensional Scaling of Large Data". *Proceedings of the 14th International Symposium on Graph Drawing*, 42–53. 2007.

[39] Bruce, L., C. Koger, and J. Li. "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction". *IEEE Transactions on Geoscience and Remote Sensing*, 40(10):2331–2338, Oct 2002.

[40] Burer, S. and A. Letchford. "Non-convex mixed-integer nonlinear programming: A survey". *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012.

[41] Burk, I. "Fast and efficient spectral clustering". Website, Sep 2012. URL http://www.mathworks.com/matlabcentral/fileexchange/34112-fast-and-efficient-spectral-clustering.

[42] Cai, S. "Hyperspectral Imagery Visualization Using Double Layers". *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3028–3036, Oct 2007.

[43] Cha, S. "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions". *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007.

[44] Chan, W. *A survey on Multivariate Data visualization*. Technical report, Hong King University of Science and Technology, Department of Computer Science and Engineering, 2006.

[45] Chang, C. "Spectral information divergence for hyperspectra; image analysis". *Proc. Geoscience and Remote Sensing Symposium*, 1:506–511, 1999.

[46] Chang, C. "Anomaly detection and classification for hyperspectral imagery". *IEEE Transactions on Geoscience and Remote Sensing*, 40(6):1314–1325, Jun 2002.

[47] Chang, C. "Constrained Band Selection for Hyperspectral Imagery". *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1575–1585, Jun 2006.

[48] Chang, K. "Principal curve classifier-a nonlinear approach to pattern classification". *IEEE World Congress on Computational Intelligence*, volume 1, 695–700. Anchorage, AK, May 1998.

[49] Chang, W., C. Lee, and C. Lin. *A Revisit to Support Vector Data Description (SVDD)*. Technical report, National Taiwan University, 2013.

[50] Chen, G. and S. Qian. "Dimensionality reduction of hyperspectral imagery using improved locally linear embedding". *Journal of Applied Remote Sensing*, 1(013509):1–10, 2007.

[51] Chen, J., H. Fang, and Y. Saad. "Fast Approximate kNN Graph Construction for High Dimensional Data via Recursive Lanczos Bisection". *Journal of Machine Learning Research*, 10:1989–2012, Sep 2009.

[52] Chen, X. and D. Cai. "Large Scale Spectral Clustering with Landmark-Based Representation". *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 313–318. 2011.

[53] Chen, Y., C. Qu, and Z. Lin. "Supervised Locally Linear Embedding Based Dimension Reduction for Hyperspectral Image Classification". *IEEE International Geoscience and Remote Sesning Symposium*, 3578–3581. Melbourne, VIC, July 2013.

[54] Cheriyadat, A. and L. Bruce. "Why Principal Component Analysis is not an Appropriate Feature Extraction Method for Hyperspectral Data". *Proceedings of the International Geoscience and Remote Sensing Symposium*, volume 6, 3420–3422. Toulouse, France, July 2003.

[55] Chiang, S., C. Chang, and I. Ginsberg. "Unsupervised target detection in hyperspectral images using projection pursuit". *IEEE Transactions on Geoscience and Remote Sensing*, 39(7):1380–1391, Jul 2001.

[56] Chiu, P. and C.L. Bloebaum. "Hyper-Radial Visualization for decision-making in Multi-Objective Optimization". *46th AIAA Aerospace Sciences Meeting and Exhibit*. Reno, Nevada, January 2008.

[57] Chu, C., I. Tsang, and J. Kwok. "Scaling up support vector data description by using core sets". *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks*. 2004.

[58] Chunhui, Z., W. Yulei, and M. Feng. "Kernel ICA feature extraction for anomaly detection in hyperspectral imagery". *Chinese Journal of Electronics*, 21(2):265–269, Apr 2012.

[59] Comrey, A. and H. Lee. *A first course in factor analysis*. Psychology Press, 2nd edition, 2013.

[60] Cortez, P., A. Cerdeira, F. Almeida, T. Matos, and J. Reis. "Modeling wine preferences by data mining from physicochemical properties". *Decision Support Systems, Elsevier*, 47(4):547–553, 2009.

[61] Cox, T. and M. Cox. *Multidimensional Scaling*. CRC Press, 2 edition, 2000.

[62] Dasgupta, S. and A. Gupta. "An elementary proof of a theorem of Johnson and Lindenstrauss". *Random Structures & Algorithms*, 22(1):60–65, 2003.

[63] Datta, A., S. Ghosh, and A. Ghosh. "Band Elimination of Hyperspectral Imagery Using Correlation of Partitioned Band Images". *International Conference on Advances in Computing, Communications, and Informatics*, 412–417. Aug 2013.

[64] Devroye, N. "Statistical Detection Theory Lecture Notes", Spring 2010. Http://www.ece.uic.edu/ devroye/courses/ECE531/lectures/b2c3.pdf.

[65] Ding, C. and X. He. "K-means Clustering via Principal Component Analysis". *Proceedings of the twenty-first international conference on Machine Learning*, 29–38. Banff, Jul 2004.

[66] Doster, T., D. Ross, and D. Messinger amd W. Basener. "Anomaly Clustering in Hyperspectral Images". *SPIE Defense, Security, and Sensing*. SPIE, May 2009.

[67] Drineas, P and M. Mahoney. "On the Nystrm Method for Approximating a Gram Matrix for Improved Kernel-Based Learning". *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[68] Duda, R., P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, second edition, 2001.

[69] Elkan, C. "Using the Triangle Inequality to Accelerate k-means". *Twentieth International Conference on Machine Learning*, volume 3, 147–153. Washington, DC, Aug 2003.

[70] Farahat, A., A. Ghodsi, and M. Kamel. "A novel greedy algorithm for Nystrom approximation". *International Conference on Artificial Intelligence and Statistics*, 269–277. 2011.

[71] Faulconbridge, R., M. Pickering, and M. Ryan. "Unsupervised band removal leading to improved classification accuracy of hyperspectral images". *Proceedings of the 29th Australasian Computer Science Conference*, volume 48, 43–48. Australian Computer Society, Inc., Hobart, Australia, Jan 2006.

[72] Fern, X. and C. Brodley. "Random projection for high dimensional data clustering: A cluster ensemble approach". *International Conference on Machine Learning*, volume 3, 186–193. 2003.

[73] Frey, B. and D. Dueck. "Clustering by passing messages between data points". *Science*, 315:972–976, Feb 2007.

[74] Friesen, K. *Automatic Target Recognition for Hyperspectral Imagery*. Master's thesis, Air Force Institute of Technology, Mar 2011.

[75] Frontera-Pons, J., M. Mahot, J.P. Ovarlez, F. Pascal, S.K. Pang, and J. Chanussot. "A class of robust estimates for detection in hyperspectral images using elliptical distributions background". *International Geoscience and Remote Sensing Symposium, IGARSS*, 4166–4169. Munich, Jul 2012.

[76] Frontera-Pons, J., F. Pascal, and J. Ovarlez. "Adaptive non-Zero Mean Gaussian Detection and Application to Hyperspectral Imaging". Website, Apr 2014. URL http://adsabs.harvard.edu/abs/2014arXiv1404.2977F.

[77] Fukunaga, K and M. Mantock. "Nonparametric Discriminant Analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):671–678, 1983.

[78] Goldberg, H., H. Kwon, and N. Nasrabadi. "Kernel eigenspace separation transform for subspace anomaly detection in hyperspectral imagery". *IEEE Geoscience and Remote Sensing Letters*, 4(4):581–585, Oct 2007.

[79] Golub, G. and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, third edition, 1996.

[80] Green, A., M. Berman, P. Switzer, and M.D. Craig. "A Transformation for Ordering Multispectral Data in Terms of Image Quality with Implications for Noise Removal". *IEEE Transactions on Geoscience and Remote Sensing*, 26(1):65–74, Jan 1988.

[81] Griffin, M. and H. Burke. "Compensation of Hyperspectral Data for Atmospheric Effects". *Lincoln Laboratory Journal*, 14(1):29–54, 2003.

[82] Grinstein, G., M. Trutschl, and U. Cvek. "High-dimensional visualizations". *Data Mining Conference KDD Workshop*, 7–19. ACM Press, New York, 2001.

[83] Gu, Q., Z. Li, and J. Han. "Generalized fisher score for feature selection". Website, 2012. URL arxiv.org/pdf/1202.3725.

[84] Gu, Y., Y. Liu, and Y. Zhang. "A selective KPCA algorithm based on high-order statistics for anomaly detection in hyperspectral imagery". *IEEE Transactions on Geoscience and Remote Sensing Letters*, 1:43–47, 2008.

[85] Gunter, S., N. Schraudolph, and S. Vishwanathan. "Fast iterative kernel component analysis". *Journal of Machine Learning Research*, 8:1893–1918, 2007.

[86] Guoen, X. and S. Peiji. "Factor Analysis Algorithm with Mercer Kernel". *IEEE International Symposium on Intelligent Information Technology and Security Informatics*, 202–205. 2009.

[87] Gupta, M. and N. Jacobson. "Wavelet principal component analysis and its application to hyperspectral images". *2006 IEEE International Conference on Image Processing*, 1–4. 2006.

[88] Gurram, P and H. Kwon. "Support-vector-based hyperspectral anomaly detection using optimized kernel parameters". *IEEE Geoscience and Remote Sensing Letters*, 8(6):1060–1064, 2011.

[89] Gurram, P., H. Kwon, and T. Han. "Sparse kernel-based hyperspectral anomaly detection". *IEEE Geoscience and Remote Sensing Letters*, 9(5):943–947, 2012.

[90] Guyon, I., S. Gunn, A. Ben-Hur, and G. Dror. "Result Analysis of the NIPS 2003 Feature Selection Challenge". *Advances in Neural Information Processing Systems*, 17:545–552, 2004.

[91] Guyon, I., S. Gunn, M. Nikravesh, and L.A. Zadeh. "Feature Extraction Foundations and Applications: Extra Materials". Website, 2006. URL http://extras.springer.com/2006/978-3-540-35487-1/Data/ARCENE.

[92] Halko, N., P. Martinsson, and J. Tropp. "FINDING STRUCTURE WITH RANDOMNESS: PROBABILISTIC ALGORITHMS FOR CONSTRUCTING APPROXIMATE MATRIX DECOMPOSITIONS". *SIAM Review*, 53(2):217–288, 2011.

[93] Han, J., J.C. Rodriguez, and M. Beheshti. "Discovering Decision Tree Based Diabetes Prediction Model". *Adavnces in Software Engineering, Communications in Computer and Information Science*, 30:99–109, 2009.

[94] Harsanyi, J.C. *Detection and classification of sub-pixel spectral signatures in hyperspectral image sequences*. Ph.D. thesis, Univ. Maryland-Baltimore County, 1993.

[95] Hastie, T. and W. Stuetzle. "Principal Curves". *Journal of the American Statistical Association*, 84(406):502–516, June 1989.

[96] He, X., D. Cai, S. Yan, and H.J. Zhang. "Neighborhood Preserving Embedding". *Tenth IEEE International Conference on Computer Vision*, volume 2, 1208–1213. Beijing, Oct 2005.

[97] Hoffman, H. "Kernel PCA for novelty detection". *Pattern Recognition*, 40:863–874, 2007.

[98] Hoffman, P. and G. Grinstein. "Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations". *NPIV Workshop on New Paradigms in Information Visualization and Manipulation*. 1999.

[99] Hourdakis, N., M. Argyriou, E. Petrakis, and E. Milios. "Hierarchical Clustering in Medical Document Collections: the BIC-Means Method". *Journal of Digital Information Management*, 8(2):71–77, 2010.

[100] Hsueh, M. and C. Chang. "Adaptive causal anomaly detection for hyperspectral imagery". *Geoscience and Remote Sensing Symposium, IGARSS '04 Proceedings*, 5:3222–3224, Sep 2004.

[101] Hua, X and S Ding. "Incremental learning algorithm for support vector data description". *Journal of Software*, 6(7):1166–1172, 2011.

[102] Hwang, W. and K. Wen. "Fast kNN classification algorithm based on partial distance search". *Electronics Letters*, 34(21):2062–2063, Oct 1998.

[103] Hyvarinen, A., J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, 2001.

[104] Hyvrinen, A. "Fast and robust fixed-point algorithms for independent component analysis". *IEEE Transactions on Neural Networks*, 10(3):626–634, May 1999.

[105] Inselberg, A. and B. Dimsdale. "Parallel coordinates: A tool for visualizing multi-dimensional geometry". *Proceedings of the First IEEE Conference on Visualization*, 361 – 378, 1990.

[106] Izenman, A. and Y. Shan. "Outlier detection using the smallest kernel principal components", 2007. Temple University: astro.temple.edu/ alan/OutlierDetectionpaper.PDF.

[107] Jablonski, J. *Reconstruction Error and Principal Component Based Anomaly Detection in Hyperspectral Imagery*. Master's thesis, Air Force Institute of Technology, Mar 2014.

[108] James, E. and S. Annadurai. "An Efficient Bayesian Approach to Face Recognition based on Wavelet Transform". *International Journal of Computer Applications*, 15(8):22–26, Feb 2011.

[109] Jennrich, R. "A simple general procedure for orthogonal rotation". *Psychometrika*, 66(2):289–306, Jun 2001.

[110] Johnson, R. *Improved feature extraction, feature selection, and identification techniques that create a fast unsupervised hyperspectral target detection algorithm*. Master's thesis, Air Force Institute of Technology, Mar 2007.

[111] Johnson, R., J. Williams, and K. Bauer. "AutoGAD: An improved ICA-based hyperspectral anomaly detection algorithm". *IEEE Transactions on Geoscience and Remote Sensing*, 51(6):3492–3503.

[112] Johnson, W.B. and J. Lindenstrauss. "Extensions of Lipschitz maps into a Hilbert space". *Contemporary Mathematics*, 26:189–206, 1984.

[113] Jordan, M. "CS 281B/Stat241B: Advanced Topics in Learning & Decision Making Lecture Notes", Spring 2004. Http://www.cs.berkeley.edu/ jordan/courses/281B-spring04/lectures/lec3.pdf.

[114] Jr., O. Carvalho and P. Meneses. "Spectral correlation mapper: an improvement on the spectral angle mapper (SAM)". *Proc. 9th Airborne Earth Sci. Workshop*, 2000.

[115] Kaiser, H. "The varimax criterion for analytic rotation in factor analysis". *Psychometrika*, 23(3):187–200, 1958.

[116] Kao, Y. and S. Lee. "Combining K-means and particle swarm optimization for dynamic data clustering problems". *IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 1, 757–761. IEEE, 2009.

[117] Keahey, T. A. "Visualization of high-dimensional clusters using nonlinear magnification". *Electronic Imaging'99, International Society for Optics and Photonics*, 228–235, 1999.

[118] Kehrer, J. and H. Hauser. "Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey". *IEEE Trans. on Visualization and Computer Graphics*, 19(3):495–513, Mar 2013.

[119] Keim, D. "Information Visualization and Visual Data Mining". *IEEE Trans. on Visualization and Computer Graphics*, 7(1):100–107, 2002.

[120] Keshava, N. "Best Band Selection for Detection in Hyperspectral Processing". *International Conference on Acoustics, Speech, and Signal Processing*, volume 5, 3149–3152. Salt Lake City, UT, May 2001.

[121] Khazai, S., S. Homayouni, A. Safari, and B. Mojaradi. "Anomaly detection in hyperspectral images based on an adaptive support vector method". *IEEE Geoscience and Remote Sensing Letters*, 8(2):646–650, 2011.

[122] Khazai, S., A. Safarai, B. Mojaradu, and S. Homayouni. "Improving the SVDD approach to hyperspectral image classification". *IEEE Geoscience and Remote Sensing Letters*, 9(4):594–598, 2012.

[123] Kim, D. and L. FInkel. "Hyperspectral Image Processing Using Locally Linear Embedding". *IEEE EMBS Conference on Neural Engineering*, 316–319. Capri Island, Italy, March 2003.

[124] Kim, K, M. Franz, and B. Schlkopf. "Iterative kernel principal component analysis for image modeling". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1351–1366, 2005.

[125] Kim, S., A. Magnani, and S. Boyd. "Optimal kernel selection in kernel fisher discriminant analysis". *Proceedings of the 23rd international conference on Machine learning*, 465–472. ACM, 2006.

[126] Kobayashi, H., K. Misue, and J. Tanaka. "Colored Mosaic Matrix: Visualization Technique for High-Dimensional Data". *Information Visualization, 17th International Conference*, 378–383. Londo, UK, July 2013.

[127] Kotwal, K. and S. Chaudhuri. "An Optimization-Based Approach to Fusion of Hyperspectral Images". *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):501–509, 2012.

[128] Kouropteva, O., O. Okun, and M. Pietikainen. "Incremental locally linear embedding algorithm". *Image Analysis: 14th Scandinavian Conference*, 521–530. SCIA, Joensuu, Finland, June 2005.

[129] Kromesch, S. and S. Juhasz. "High Dimensional Data Visualization". *Proc. International Symposium of Hungarian Researchers on Computational Intelligence*, 230–237. Budapest, HU, Nov 2005.

[130] Kuang, D. "Accelerate Matlab K-means with Simple Patches". Website, March 2014. URL http://www.cc.gatech.edu/grads/d/dkuang3/software/kmeans3.html.

[131] Kwon, H., S. Der, and N. Nasrabadi. "Adaptive anomaly detection using subspace separation for hyperspectral imagery". *Optical Engineering*, 42(11):3342–3351, 2003.

[132] Kwon, H. and P. Gurram. "Optimal kernel bandwidth estimation for hyperspectral kernel-based anomaly detection". *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2812–2815. 2010.

[133] Kwon, H. and N. Nasrabadi. "Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery". *IEEE Transactions on Geoscience and Remote Sensing*, 43(2):388–397, Feb 2005.

[134] Lal, T., O. Chapelle, and B. Scholkpf. *Combining a Filter Method with SVMs*, 439–445. Springer Berlin Heidelberg, 2006.

[135] Lappalainen, H. and A. Honkela. *Advances in independent component analysis*, chapter Bayesian non-linear independent component analysis by multi-layer perceptrons, 93–121. Springer London, 2000.

[136] Lavanya, A. and S. Sanjeevi. "An Improved Band Selection Technique for Hyperspectral Data Using Factor Analysis". *Journal of the Indian Society of Remote Sensing*, 41(2):199–211, Jun 2013.

[137] Lee, C. and D. Landgrebe. "Analyzing High-Dimensional Multispectral Data". *IEEE Transactions on Geoscience and Remote Sensing*, 31(4):792–800, 1993.

[138] Li, J., S. Chu, J. Pan, and J. Ho. "A Novel Matrix Norm Based Gaussian Kernel for Feature Extraction of Images". *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. Pasadena, CA, Dec 2006.

[139] Li, M., J. Kwok, and B. Lu. "Making Large-Scale Nystrom Approximation Possible". *Porceedings of the 27th International COnference on Machine Learning*. 2010.

[140] Li, P., T. Hastie, and K.W. Church. "Very sparse random projections". *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*, 287–296. ACM, 2006.

[141] Li, W., S. Prasad, and J. Fowler. "Classification and Reconstruction From Random Projections for Hyperspectral Imagery". *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):833–843, Feb 2013.

[142] Liang, Z. and Y. Lee. "Eigen-analysis of nonlinear PCA with polynomial kernels". *Statistical Analysis and Data Mining*, 6(6):529–544, 2013.

[143] Liu, C. "Gabor-Based Kernel PCA with Fractional Power Polynomial Models for Face Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):572–581, May 2004.

[144] Liu, W. and C. Chang. "A nested spatial window-based approach to target detection for hyperspectral imagery". *IEEE*, 1(20):264–268, 2004.

[145] Liu, Z., Y. Gu, and Y. Zhang. "Comparative Analysis of Feature Extraction Algorithms with Different Rules for Hyperspectral Anomaly Detection". *First International Conference on Pervasive Computing, Signal Processing, and Applications*. IEEE Computer Society, Harbin Institute of Technology, China, Sep 2010.

[146] Lu, J., K.N. Plataniotis, and A.N. Venetsanopoulos. *Support Vector Machines: Theory and Applications*, chapter Kernel discriminant learning with application to face recognition, 275–296. Springer, Berlin Heidelberg, 2005.

[147] Lu, T. "Robust Locally Linear Embedding and Application in High Dimensional Data". *Fifth International Conference on Natural Computation*, volume 1, 299–306. Tianjin, Aug 2009.

[148] van der Maaten, L. and G. Hinton. "t-Distributed Stochastic Neighbor Embedding Implementations". Website, Aug 2014. URL http://homepage.tudelft.nl/19j49/t-SNE.html.

[149] van der Maaten, LGP, EO Postma, and HJ van den Herik. *DImensionality Reduction: a Comparative Review*. Technical Report 2009-005, Tilburg University, 2009.

[150] van der Maaten, L.J. P. and G.E. Hinton. "Visualizing High-Dimensional Data Using t-SNE". *Journal of Machine Learning Research*, 9:2579–2605, Nov 2008.

[151] Maathuis, M. "Multivariate Statistics Lecture Notes", Fall 2008. Http://stat.ethz.ch/ mmarloes/teaching/fall08/Notes5.pdf.

[152] Mallapragada, P., R. Jin, A. Jain, and Y. Li. "SemiBoost: boosting for semi-supervised learning". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2000–2014, 2009.

[153] Mallat, S. *A Wavelet Tour of Signal Processing The Sparse Way*. Elsevier, 3 edition, 2009.

[154] Manolakis, D., R. Lockwood, T. Cooley, and J. Jacobson. "Is there a best hyperspectral detection algorithm?" *Proceedings of SPIE*, 7334(1), 2009.

[155] Martinez-Uso, A., F. Pla, J.M. Sotoca, and P. Garcia-Sevilla. "Clustering-Based Hyperspectral Band Selection Using Information Measures". *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4158–4171, 2007.

[156] van der Meer, F. "The effectiveness of spectral similarity meaures for the Analysis of hyperspectral imagery". *International Journal of Applied Earth Observation and Geoinformation*, 8(1):3–17, January 2006.

[157] van der Meer, F. and S.M. Jong. *Imaging Spectrometry: Basic Principles and Prospective Applications*, volume XXIII. Springer, 2002.

[158] Messinger, D., A. Ziemann, A. Schlamm, and B. Basener. "Spectral Image Complexity Estimated Through Local Convex Hull Volume". *2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*. IEEE, Reykjavik, Iceland, Jun 2010.

[159] Mika, S., G. Ratsch, J. Weston, B. Scholkopf, A. Smola, and K.R. Muller. "Constructing Descriptive and Discriminative Nonlinear Features: Rayleigh Coefficients in Kernel Feature Spaces". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):623–628, May 2003.

[160] Miller, M. *Eploitation of Intra-Spectral Band Correlation for Rapid Feature Selection, and Target Identification in Hyperspectral Imagery*. Master's thesis, Air Force Institute of Technology, Mar 2010.

[161] Minh, H., P. Niyogi, and Y. Yao. "Mercers Theorem, Feature Maps, and Smoothing". *Learning Theory, Lecture Notes in Computer Science*, volume 4005, 154–168. 2006.

[162] Mohamed, A. and R. Yampolskiy. "Using Discrete Wavelet Transform and Eigenfaces for Recognizing Avatars Faces". *17th International Conference on Computer Games*. Louisville, KY, July 2012.

[163] Mukundan, R., S.H. Ong, and P.A. Lee. "Image Analysis by Tchebichef Moments". *IEEE Transactions on Image Processing*, 10(9):1357–1364, Sep 2001.

[164] Naim, A.M., P.W. Chiu, C. Bloebaum, and K. Lewis. "Hyper-Radial Visualization for Multi-Objective Decision-making Support Under Uncertainty Using Preference Ranges: The PRUF Method". *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Victoria, BC, CA, September 2008.

[165] Nasrabadi, N. "Kernel subspace-based anomaly detection for hyperspectral imagery". *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, WHISPERS*, 1–4. Aug 2009.

[166] Nene, S. and S. Nayar. "A Simple Algorithm for Nearest Neighbor Search in High Dimensions". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):989–1003, Sep 1997.

[167] Ng, A., M. Jordan, and Y. Weiss. "On spectral clustreing: analysis and an algorithm". *Advances on Neural Information Processing Systems 14*, volume 2, 849–856. MIT Press, Cambridge, MA, 2002.

[168] Nielsen, A. and M. Canty. "Kernel principal component analysis for change detection". *Proc. of SPIE, Image and Signal Processing for Remote Sensing XIV*, 7109:71090T1–71090T10, 2008.

[169] Paciencia, T. *Multi-Objective Optimization of Mixed Variable, Stochastic Systems Using Single-Objective Formulations*. Master's thesis, Air Force Institute of Technology, 2008.

[170] Paiva, A., J. Xu, and J. Principe. *Independent Component Analysis and Blind Signal Separation*, chapter Kernel principal components are maximum entropy projections, 846–853. Springer Berlin Heidelberg, 2006.

[171] Park, H. and C. Jun. "A simple and fast algorithm for K-medoids clustering". *Expert Systems with Applications*, 36(2):3336–3341, Mar 2009.

[172] Pelleg, D and A. Moore. "X-means: extending k-means with efficient estimation of the number of clusters". *Proceedings of the Seventeenth International Conference on Machine Learning*, 727–734, 2000.

[173] Pu, R. and P. Gong. "Band Selection from Hyperspectral Data for Conifer Species Identification". *Geographic Information Sciences*, 6(2):137–142, Dec 2000.

[174] Rao, R. and S.K. Card. "The Table Lens: Merging GrGraphic and Symbolic Representations in an Interactive Fcous + Context Visualization for Tabular Information". *Proc. ACM CHI Conference on Human Factors in Computer Systems: Celebrating Interdependence*, 318–322. Boston, MA, 1994.

[175] Rasti, B., J. Sveinsson, M. Ulfarsson, and J. Benediktsson. "Hyperspectral Image Denoising Using 3D Wavelets". *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp1349–1352. IEEE, 2012.

[176] Reddy, S., S. Kodali, and J. Gundabathina. "Classification of Vertebral Column Using Naive Bayes Technique". *International Journal of Computer Applications*, 58(7):38–42, Nov 2012.

[177] Reed, I.S. and X. Yu. "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution". *IEEE Trans. Acoust., Speech, Signal Processing*, 38:1760–1770, Oct 1990.

[178] Ren, H., C. Chen, and H. Chen. "Weighted anomaly detection for hyperspectral remotely sensed images". *Proc. SPIE 5995, Chemical and Biological Standoff Detection III*, 599507:1760–1770, Nov 2005.

[179] de Ridder, D., O. Kouropteva, O. Okun, M. Pietikainen, and R.P. Duin. "Supervised locally linear embedding". *International Conference on Artificial Neural Networks and Neural Information Processing*, 333–341. Instanbul, 2003.

[180] Rosario, D. "A nonparametric F-distribution anomaly detector for hyperspectral imagery". *IEEE*, 5(12):2022–2029, 2005.

[181] Roweis, S. and L. Saul. "Nonlinear dimensionality reduction by locally linear embedding". *Science*, 290(5500):2323–2326, Dec 2000.

[182] Roweis, S. and L. Saul. "An Introduction to locally linear embedding", January 2001. Http://www.cs.toronto.edu/ roweis/lle/publications.html.

[183] Saunders, J., N. Morrow-Howell, E. Spitznagel, P. Dore, E.K. Proctor, and R. Pescarino. "Imputing missing data: A comparison of methods for social work researchers". *Social Work Research*, 30(1):19–31, 2006.

[184] Scholkopf, B., D. Achlioptas, and F. McSherry. "Sampling techniques for kernel methods". *Advances in Neural Information Processing Systems*, volume 1. MIT Press, 2001.

[185] Scott, D.W. "On optimal and data-based histograms". *Biometrika*, 66(3):605–610, 1979.

[186] Shalizi, C. "Statistics 36-350: Data Mining, Factor Analysis Lecture Notes", Fall 2009. Http://www.stat.cmu.edu/ cshalizi/350/lectures/12/lecture-12.pdf.

[187] Shalizi, C. "Statistics 36-350: NonlinearLectNonlinear Dimensionality Reduction I: Local Linear Embedding", Fall 2009. Http://www.stat.cmu.edu/ cshalizi/350/lectures/14/lecture-14.pdf.

[188] Shen, H., S. Jegelka, and A. Gretton. "Fast Kernel ICA using an Approximate Newton Method". *Eleventh International Conference on Artificial Intelligence and Statistics*. San Juan, Puerto Rico, Mar 2007.

[189] Shen, L. and S. Jia. "Three-Dimensional Gabor Wavelets for Pixel-Based Hyperspectral Imagery Classification". *IEEE Transactions on Geoscience and Remote Sensing*, 49(12):5039–5046, Dec 2011.

[190] Shi, J. and J. Malik. "Normalized Cuts and Image Segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[191] Smetek, T. *Hyperspectral imagery target detection using improved anomaly detection and signature matching methods*. Ph.D. thesis, Air Force Institute of Technology, Mar 2007.

[192] Smetek, T. and K. Bauer. "Finding Hyperspectral Anomalies Using Multivariate Outlier Detection". *IEEE Aerospace Conference*. IEEE, 2007.

[193] Smith, R. "Introduction to Hyperspectral Imaging". Website, January 2012. URL http://www.microimages.com/documentation/Tutorials/hyprspec.pdf.

[194] Snyder, D., J. Kerekes, I. Fairweather, R. Crabtree, J. Shive, and S. Hager. "Development of a Web-based Application to Evaluate Target Finding Algorithms". *Proceedings of the 2008 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, volume 2, 915–918. Boston, MA, 2008.

[195] Soofbaf, S., M. ValadanZoej, H. Fahimnejad, and H. Ashoori. "Efficient detection of anomalies in hyperspectral images". *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37, 2008.

[196] Sotoca, J., F. Pla, and A. Klaren. "Unsupervised band selection for multispectral images using information theory". *Proceedings of the 17th International Conference on Pattern Recognition*, volume 3, 510–513. Aug 2004.

[197] Su, T. and J. Dy. "A deterministic method for initializing k-means clustering". *16th IEEE International Conference on Tools with Artificial Intelligence*, 784–786. IEEE, 2004.

[198] Sugar, C. and G. James. "Finding the number of clusters in a dataset". *Journal of the American Statistical Association*, 98(463), 2003.

[199] Taitano, Y., B. Gaier, and K. Bauer. "A locally adaptable iterative RX detector". *EURASIP Journal on Advanced Signal Processing, Special Issue on Advanced Image Processing for Defense and Security Applications*, 2010:1–10, 2010.

[200] Takiguchi, T. and Y. Ariki. "Robust Feature Extraction Using Kernel PCA". *International Conference on Acoustics, Speech, and Signal Processing*, 1–4. IEEE, Toulouse, May 2006.

[201] Tao, Y., K. Yi, C. Sheng, and P. Kalnis. "Efficient and accurate nearest neighbor and closest pair search in high-dimensional space". *ACM Transactions on Database Systems*, 35(3):20, Jul 2010.

[202] Tavakkoli, A., M. Nicolescu, M. Nicolescu, and G. Bebis. "Incremental SVDD training: improving efficiency of background modeling in videos". *Proceedings of the 10th IASTED International Conference*, volume 623. 2008.

[203] Tax, D. and R. Duin. "Support vector data description". *Machine Learning*, 54:45–66, 2004.

[204] Thomas, A. "Extending the RX anomaly detection algorithm to continuous and spatial domains". *Proc. IEEE Southeastcon*, 557–562, Apr 2008.

[205] Torrieri, D. "The eigenspace transform for neural network classifiers". *Neural Networks*, 12(3):419–427, April 1999.

[206] Turk, M. and A. Pentland. "Eigenfaces for Recognition". *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[207] Venkatasubramanian, S. and Q. Wang. "The Johnson-Lindenstrauss transform: an empirical study". *Proc. of the thirteenth workshop on Algorithm Engineering and Experiments*, 164–173. SIAM, 2011.

[208] Vidakovic, B. *Statistical Modeling of Wavelets*. John Wiley & Sons, 1999.

[209] Wackerly, D., W. Mendenhall III, and R. Scheaffer. *Mathematical Statistics with Applications*. Ducbury Thomson Learning, 6 edition, 2002.

[210] Wang, B., X. Wang, and Z. Chen. "Spatial Entropy Based Mutual Information in Hyperspectral Band Selection for Supervised Classification". *International Journal of Numerical Analysis and Modeling*, 9(2):181–192, 2012.

[211] Wang, J., H. Lu, K.N. Plataniotis, and J. Lu. "Gaussian Kernel Optimization for Pattern Classification". *Pattern Recognition*, 42(7):1237–1247, 2009.

[212] Wang, J., K.N. Plataniotis, and A.N. Venetsanopolous. "Selecting Kernel Eigenfaces for Face Recognition with One Training Sample Per Subject". *IEEE International COnference on Multimedia and Expo*, 1637–1640. IEEE, Jul 2006.

[213] Wang, Y., C. Zao, and Y. Wang. "Anomaly detection using subspace band section based RX algorithm". *IEEE International Conference on Multimedia Technology*, 3436–3439. IEEE, 2011.

[214] Wanga, W., Z. Xua, W. Luc, and X. Zhanga. "Determination of the spread parameter in the Gaussian kernel for classification and regression". *Neurocomputing*, 55(3):643–663, 2003.

[215] Williams, C. and M. Seeger. "Using the Nystrom Method to Spped Up Kernel Machines". *14th Annual Conference on Neural Information Processing Systems*, EPFL-CONF-161322, 682–688. 2001.

[216] Williams, J. *Addressing correlation effects in real-time anomaly detection of hyperspectral imagery*. Ph.D. thesis, Air Force Institute of Technology, Sep 2012.

[217] Williams, J., T. Bihl, and K. Bauer. "Mitigation of Correlation and Heterogeneity Effects in Hyperspectral Data". *Intelligent Engineering Systems Through Artificial Neural Networks*, 20:501–507, 2010.

[218] Wu, N. and J. Zhang. "Factor Analysis Based Anomaly Detection". *IEEE Workshop on Information Assurance*. West Point, NY, June 2003.

[219] Xaio, Y., B. Liu, and L. Cao. "K-farthest-neighbors-based concept boundary determination for support vector data description". *Proceedings of the 19th ACM international conference on information and knowledge management*. 2010.

[220] Xia, D., F. Wu, X. Zhang, and Y. Zhuang. "Local and global approaches of affinity propagation clustering for large scale data". *Journal of Zhejiang University SCIENCE A*, 9(10):1373–1381, 2008.

[221] Xie, S., A. Lawniczak, S. Krishnan, and P. Lio. "Wavelet Kernel Principal Component Analysis in NoisyMultiscale Data Classification". *International Scholarly Research Network Computational Mathematics*, (197352):1–13, 2012.

[222] Xu, Y., W. Hong, X. Li, and J. Song. "Parallel Dual Visualization of Multidimensional Multivariate Data". *Proc. IEEE International Conference on Integration Technology*, 263–268. Shenzhen, CH, Mar 2007.

[223] Yang, J., A.F. Frangi, J.Y. Yang, D. Zhang, and Z. Jin. "KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):230–244, Feb 2005.

[224] Yang, M. "Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods". *5th IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, Washington, D.C., 2002.

[225] Zare, A. *Hyperspectral Endmember Detection and Band Selection Using Bayesian Methods*. Ph.D. thesis, University of Florida, 2008.

[226] Zare-Baghbidi, M. and S. Homayouni. "Fast hyperspectral anomaly detection for environmental application". *Journal of Applied Remote Sensing*, 7:1–11, 2013.

[227] Zeng, X. and T.S. Durrani. "Band Selection for Hyperspectral Images Using Copulas-based Mutual Information". *IEEE/SP 15th Workshop on Statistical Signal Processing*, 341–344. Cardiff, Aug-Sep 2009.

[228] Zhang, K. and J. Kwok. "Clustered Nystrom Method for Large Scale Manifold Learning and DImension Reduction". *IEEE Transactions on Neural Networks*, 21(10):1576–1587, Oct 2010.

[229] Zhang, Z. and L. Zhao. "Probability-based locally linear embedding for classification". *Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 3, 243–247. Haikou, China, 2007.

[230] Zhu, M. and T. Hastie. "Feature extraction for nonparametric discriminant analysis". *Journal of Computational and Graphical Statistics*, 12(1):101–120, 2003.

[231] Zhu, X., Z. Huang, H. Shen, J. Cheng, and C. Xu. "Dimensionality reduction by mixed kernel canonical correlation analysis". *Pattern Recognition*, 45:3003–3016, 2012.

[232] Zhu, Y., J. Yu, and C. Jia. "Initializing k-means clustering using affinity propagation". *Ninth International Conference on Hybrid Intelligent Systems*, volume 1. IEEE, 2009.

[233] Ziemann, A., D. Messinger, and J. Albano. "Target detection performed on manifold approximations recovered from hyperspectral data". S. Shen and P. Lewis (editors), *SPIE Defense, Security, and Sensing*, volume 8743. SPIE, 2013.

Vita

Maj Todd Paciencia graduated from Kenmore East High School in Tonawanda, NY in 2000. He graduated from Binghamton University in Binghamton, NY in May 2003 with a Bachelor of Arts in Mathematical Sciences and was commissioned through Air Force Officer Training School in February 2004.

Maj Paciencia served his first assignment as a Lead Analyst for the Joint GPS Combat Effectiveness Joint Test & Evaluation. Upon completion, he became the Analysis Branch Chief of the Air Force Joint Test & Evaluation Group Quick Reaction Test Force, and following, worked for both AFOTEC, as a Systems Analyst, and AFRL, as a member of one of its SPACE-CHOP missions.

In 2008, he completed his Master's of Science in Operations Research from the Air Force Institute of Technology in-residence. Shortly thereafter, while assigned to Rome Labs as an Advanced Computing Research Engineer, he deployed to Multi-National Forces Iraq CJ9 and served as a Strategic Assessment Analyst. Returning to Rome Labs, Maj Paciencia served as the Squadron Section Commander. Following, he was assigned to the 607 AOC in Korea as the Operational Assessment Team Chief.

In January 2012, he entered the Air Force Institute of Technology for PhD studies. Upon graduation, he will be assigned to Headquarters AF A9.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE (DD–MM–YYYY)	2. REPORT TYPE	3. DATES COVERED *(From — To)*
26-12-2014	Doctoral Dissertation	Jan 2012-Dec 2014

4. TITLE AND SUBTITLE
Improving Non-Linear Approaches to Anomaly Detection, Class Separation, and Visualization

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Paciencia, Todd J., Major, USAF

5d. PROJECT NUMBER

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
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT-ENS-DS-14-D-15

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

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

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

12. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution Statement A:
Approved For Public Release; Distribution Unlimited

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

14. ABSTRACT

Linear approaches for multivariate data analysis are popular due to their lower complexity, reduced computational time, and easier interpretation. In many cases, linear approaches produce adequate results; however, non-linear methods may generate more robust transformations, features, and decision boundaries. Of course, these non-linear methods present their own unique challenges that often inhibit their use.

In this research, improvements to existing non-linear techniques are investigated for the purposes of providing better, timely class separation and improved anomaly detection on various multivariate datasets, culminating in application to anomaly detection in hyperspectral imagery. Primarily, kernel-based methods are investigated, with some consideration towards other methods. Improvements to existing linear-based algorithms are also explored. Here, it is assumed that classes in the data have minimal overlap in the originating space or can be made to have minimal overlap in a transformed space, and that class information is unknown *a priori*. Further, improvements are demonstrated for global anomaly detection on a variety of hyperspectral imagery, utilizing fusion of spatial and spectral information, factor analysis, clustering, and screening. Additionally, new approaches for *n*-dimensional visualization of data and decision boundaries are developed.

15. SUBJECT TERMS
Anomaly detection, non-linear class separation, hyperspectral, kernel methods, visualization

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Kenneth W. Bauer, Jr., AFIT/ENS
U	U	U	UU	354	19b. TELEPHONE NUMBER (Include Area Code) (937)255-3636x4328 kenneth.bauer@afit.edu

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