Thesis Title: False Alarm Mitigation in Hyperspectral Detection of Gaseous Chemicals using Knowledge of Chemical Library and Residuals

Author: Andrew Lai

Department: Electrical and Computer Engineering

Approved for Thesis Requirements of the Master of Science Degree

Thesis Adviser: Dr. Dimitris G. Manolakis

Thesis Adviser: Dr. Vinay K. Ingle

Thesis Reader: Dr. Steven Golowich

Thesis Reader: Dr. Charles DiMarzio

Department Chair: Prof. Ali Abur

Graduate School Notified of Acceptance:

Director of the Graduate School: Prof. Sara Wadia-Fascetti
FALSE ALARM MITIGATION IN HYPERSPECTRAL DETECTION OF GASEOUS CHEMICALS USING KNOWLEDGE OF CHEMICAL LIBRARY AND RESIDUALS

A Thesis Presented
by
Andrew Lai
to
The Department of Electrical and Computer Engineering
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical and Computer Engineering

Northeastern University
Boston, Massachusetts

December 2012
Abstract

Remote sensing of chemical vapors with hyperspectral imaging devices plays an important part in many civilian and military applications. The EPA uses hyperspectral imaging to monitor the chemical vapor output from smoke stacks while the military uses it for detection of chemical warfare agents (CWAs) in the battlefield. Detection algorithms have been studied for decades now but a major limitation thus far has been the library of signatures available. In previous years, the signature libraries have been of very good quality but because of the effort needed per signature, only a handful of chemical plume signatures were available. On the other hand, reducing the resolution allowed for having a much more expansive set of signatures, but the quality of each signature becomes significantly worse. In recent years, the technology to create high resolution chemical vapor signatures have become available, increasing the size and quality of the hyperspectral signature libraries.

With the advent of a comprehensive, high quality hyperspectral chemical signature library, we can now study the chemicals themselves. This thesis presents a study of the chemical vapor signature library. The main objective of studying the library is to determine the discriminability of the signatures. We attempt to form clusters of signatures in order to see if there are inherent similarities between signatures. If clusters form naturally, then we can expect signatures within a cluster to be easily
mistaken for one another. We use two methods to cluster the library: K-means clustering and hierarchical tree-based clustering. With K-means clustering, we utilize a binomial search in order to determine the number of cluster centers given a spectral angle threshold. For tree-based clustering, we analyze the advantages and disadvantages of using the single, average and complete linkage methods for our clustering study.

We also utilize the knowledge of the library to study False Alarm Mitigation (FAM) by comparing detection results with each chemical’s nearest confusers in terms of Spectral Angle. We embed both a plume along with it’s confusers and compare detection results. The Adaptive Coherent/Cosine Estimator (ACE) detector, which is the basis detector for this study, can give both plume and confuser very similar scores. Therefore, we utilize information found in the residuals in order to separate false alarms from true positives. We also use images with real data and real background-based false positives. We later conclude that, due to the nature and origin of these false positives as compared to embedded false positives, metrics which work for one case in general does not work for the other.
Acknowledgements

The author would like to thank Professor Vinay Ingle in extending the opportunity to complete this thesis at Lincoln Lab, where he was introduced to such interesting, intelligent and kind people. I would especially like to thank Dr. Dimitris Manolakis for all his guidance and support in the direction and completion of this work. I would also like to thank Dr. Steven Golowich, Robert DiPietro, Joseph Costa and Eric Larrieux for all the conversations we’ve had covering the details of my thesis work and in their tireless efforts in getting me up to speed.

I would like to thank my parents, especially my father, for always believing in me and supporting me throughout my lifetime. You’ve taught me the importance of hard work, attention to detail, math and science in general. You’ve always allowed me to explore my interests in science and electronics.

Finally, I want to thank all my professors from Northeastern University. If it wasn’t for all of you teaching me all that I know and guiding my thirst for knowledge, I would not be as focused as I am today. I would not be where I am now without all of your help and your tireless efforts to explain all that you did in a way which captured my interests.
3 Study of Chemical Spectral Library

3.1 Overview .................................................. 18
3.2 The PNNL Spectral Library .............................. 19
  3.2.1 Downsampling ....................................... 25
3.3 Clustering ................................................. 28
  3.3.1 Overview ............................................. 28
  3.3.2 K-Means Clustering .................................. 28
  3.3.3 Hierarchical Clustering by Trees ................. 30
  3.3.4 Concerns about Clustering Methods ............... 38
  3.3.5 Modified K-means Clustering ................. 40
3.4 Effects of Spectral Resolution on Clustering .......... 44
  3.4.1 Objectives .......................................... 44
  3.4.2 Results ............................................. 44
3.5 Noise Study ............................................. 49
  3.5.1 Objectives .......................................... 49
  3.5.2 Adding Realistic Noise ......................... 49
  3.5.3 Adding Noise based on Noise Power .......... 51
  3.5.4 Confusion Matrices ............................... 56
3.6 Transmission Coefficient Study ......................... 59
  3.6.1 Minimum Detectable Amount ..................... 59
3.7 Orthogonality Study ...................................... 63
3.8 Summary ................................................. 66

4 False Alarm Analysis ........................................ 68

4.1 Least Squares Unmixing .................................. 69
4.2 Model Validation ......................................... 71
List of Figures

1.1 Some radiance data from a Hyperspectral Imager. .......................... 2

2.1 The three layer radiance model. ............................................. 6

2.2 Diagram of ACE. Plume is in red, where ACE measures the angle (instead of distance, as in Matched Filtering) .............................. 17

3.1 Histogram of the SAM of all signatures in their pure form. From the exponential rise of the curve, we can see that most signatures have a greater than 50° angle between them. Graph taken at 68 bands. .... 22

3.2 Plot of Acetoin.25T. An example of a chemical signature with 1 major spike and several smaller features. ................................. 23

3.3 Plot of HCFC142B.25T. An example of a chemical signature whose features are changing rapidly, causing sharp spikes even after down-sampling. ................................................................. 23

3.4 Plot of EDB.25T. An example of a signature with only one major feature. 24

3.5 Conversion of BUTENE-1 from high-resolution to downsampled signature used for this study. Downsampled to 68 bands. ............... 26

3.6 Plot of the high resolution and low resolution downsampled signatures for Acetoin.25T. ................................................................. 27
3.7 Signatures for DIOFORM_25T and T12DCE_25T. Based on the spectral angle, these two signatures are very similar, with an angle of only 1.88°, even though they have different magnitudes.

3.8 Cluster formed by Hierarchical tree-based methods with Complete linkage and a spectral angle threshold of 35°.

3.9 Cluster formed by Hierarchical tree-based methods with Average linkage and a spectral angle threshold of 35°.

3.10 Hierarchical Tree using the Single Linkage method.

3.11 Hierarchical Tree using the Average Linkage method.

3.12 Hierarchical Tree using the Complete Linkage method.

3.13 SAM of unclustered library.

3.14 SAM of clustered library using Single linkage method.

3.15 SAM of clustered library using Average linkage method.

3.16 SAM of clustered library using Complete linkage method.

3.17 Plot of Angular Threshold vs. Number of Clusters.

3.18 Dendrogram of Test Case Signatures, Single Linkage.

3.19 Dendrogram of Test Case Signatures, Average Linkage.

3.20 Dendrogram of Test Case Signatures, Complete Linkage.

3.21 Diagram of modified K-Means Algorithm.

3.22 Angular Threshold vs. Number of Signatures Unclustered using the Modified K-means Algorithm.

3.23 Histogram of the SAM created by comparing the threat library vs the non-threat library using the Modified K-means Algorithm.

3.24 Histograms of SAM from 8 bands to 512 bands by powers of 2.

3.25 Number of Clusters vs Angular threshold for all bands studied.
3.26 CDF of the spectral angle map taken at 5, 10, 20, etc. bands as indicated above. Lines indicate where 50% of the signature-pair angles lie. Vertical lines correspond to 51°, 65°, 71° and 75°.

3.27 CDF of the spectral angle map taken at 5, 10, 20, etc. bands as indicated above. Lines indicate where 10% of the signature-pair angles lie. Vertical lines correspond to 22°, 36°, 43° and 48°.

3.28 Plot which shows how many signatures-pair angles are below a threshold. For example, all signature-pair angles are above 1°, while almost no signature-pair angles are above 120°. The bands at which the spectral angle are calculated are indicated in the legend.

3.29 Plot of TEP.25T without noise.

3.30 Plot of TEP.25T with additive noise power at 1E-6.

3.31 Histograms of Signature Power separated as signatures with power greater than $1 \times 10^{-6}$ and those with power under this threshold.

3.32 SAM and Histogram of entire library, noise power = 0.

3.33 SAM and Histogram of entire library, noise power = $1 \times 10^{-18}$.

3.34 SAM and Histogram of entire library, noise power = $1 \times 10^{-9}$.

3.35 SAM and Histogram of entire library, noise power = $1 \times 10^{-5}$.

3.36 Histogram of the SAM of Pure Noise.

3.37 Plot of DODEC.25T, the most commonly predicted signature.

3.38 CDF of Spectral Angles between transmission coefficients sampled at varying concentration pathlengths. The values in the legend correspond to the various CL values.

3.39 Minimum Detectable Amount.
3.40 Cluster of normalized Transmission Coefficients with concentration pathlength of 10 ppm. ................................. 62
3.41 Cluster of normalized Transmission Coefficients with concentration pathlength of 1110 ppm. ................................. 62
3.42 A plot of F134A_25T showing the actual signature and the converted signature used for Hamming distance calculations. ...................... 64
3.43 Plot of CLFORM_25T, an example of a signature with only 1 sharp spike. ......................................................... 65
3.44 Plot of ET3TOL_25T, an example of a signature with several spikes. . 65
3.45 Histogram of the correlation coefficients between the Spectral Angle and the Hamming Distance. .............................. 66

4.1 Diagram showing effects of whitening and outlining what ACE and MF is composed of. We can see that the target spectra is simply the background spectra shifted over. This shifting is possible because of the linearization approximations we’ve included in the model. After demeaning and whitening, we are in the space shown to the right of the figure. MF looks at the projections of the test spectra on to the signature spectra, whereas ACE looks at the angle between the two. . 72
4.2 Embedded Data: Studentized Residuals for Chemical 1 and Chemical 2. Clearly, the studentized residuals for plume pixels are nearly white, whereas the residuals are highly correlated for the confusor. . . . . . 75
4.3 Embedded Data: Variances of Studentized Residuals. Very difficult to separate due to studentizing, which forces the variance to unity. . . . 75
4.4 Unwhitened and Whitened signatures for Chemical 1 and Chemical 2, Confusor #2 based on Mahalanobis Cosine angles (\( \angle = 32.0810^\circ \)) . . 78
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>Unwhitened and Whitened signatures for Chemical 1 and Chemical 3, Confusor #5 based on Mahalanobis Cosine angles ($\angle = 47.7908^\circ$)</td>
</tr>
<tr>
<td>4.6</td>
<td>Whitened Chemical 1 signature with indication of where the influential bands are located.</td>
</tr>
<tr>
<td>4.7</td>
<td>Embedded Background Image, Embedding Mask where red is Plume (Chemical 1) and light blue is Confusor (Chemical 2), ACE Detection Map for Embedded Chemical 1 and Chemical 2</td>
</tr>
<tr>
<td>4.8</td>
<td>RSS&lt;sub&gt;st-inf&lt;/sub&gt; vs ACE for Figure 4.7</td>
</tr>
<tr>
<td>4.9</td>
<td>Embedded Background Image, Embedding Mask where red is Plume (Chemical 1) and light blue is Confusor (Chemical 3), ACE Detection Map for Embedded Chemical 1 and Chemical 3</td>
</tr>
<tr>
<td>4.10</td>
<td>RSS&lt;sub&gt;st-inf&lt;/sub&gt; vs ACE for Figure 4.9</td>
</tr>
<tr>
<td>4.11</td>
<td>Embedded Data. Radiance vs Raw Residuals. Shows how Raw Residuals of false positives display more variation than true positives</td>
</tr>
<tr>
<td>4.12</td>
<td>Embedded Data. ACE vs SSE and ACE vs Variance of Raw Residuals. We can see that the variance of raw residuals is a scaled version of SSE.</td>
</tr>
<tr>
<td>4.13</td>
<td>Embedded Chemical 1 and Chemical 2. ACE vs RSS. We can see RSS is generally larger for False Positives (Chemical 2) compared to True Positives (Chemical 1)</td>
</tr>
<tr>
<td>4.14</td>
<td>Embedded Chemical 1 (true positive) and Chemical 2 (confusor) vs the estimated radiance.</td>
</tr>
<tr>
<td>4.15</td>
<td>Embedded Data: Chemical 1 vs Chemical 2 (confusor #2). Comparison of Variance of Raw Residuals, SoF F-Test, RSS, and RSS&lt;sub&gt;st-inf&lt;/sub&gt; as metrics.</td>
</tr>
</tbody>
</table>
4.16 Embedded Data: Chemical 1 vs Chemical (confusor #2). Detection Masks after thresholding. Threshold set to capture 99% of plume pixels.

4.17 Embedded Data: Chemical 1 vs Chemical 3 (confusor #5). Comparison of Variance of Raw Residuals, \( SoF \) F-Test, RSS, and RSS\(_{st{-inf}}\) as metrics.

4.18 Embedded Data: Chemical 1 vs Chemical 3 (confusor #5). Detection Masks after thresholding. Threshold set to capture 99% of plume pixels.

4.19 Unwhitened and Whitened signatures for Chemical 4 and Chemical 5, Chemical 4's closest confusor based on Mahalanobis Angle (\( \angle = 29.1294^\circ \)).

4.20 Embedded Data: Chemical 4 vs Chemical 5 (confusor #1). Shows the Embedded Image, ACE Detection Mask and Embedding Mask. In Embedding Mask, Red is Plume (Chemical 4) and light blue is Confusor (Chemical 5).

4.21 Embedded Data: Comparing all metrics for embedding Chemical 4 with Confusor #1 (Chemical 5) from Figure 4.20.

4.22 Embedded Data: Using data from Figure 4.21, we generated masks showing where "plume" is after thresholding. SSE can remove all of the false alarms, whereas \( SoF \) couldn’t remove any of the false alarms. Thresholding based on detecting 99% of the plume pixels.

4.23 Real Data: Chemical 1 release with ACE map. If we set a low ACE threshold, we can guarantee to detect all of the plume but also have a lot of false positives. A higher ACE threshold would miss some of the plume.

4.24 Real Data: Comparing all metrics for Chemical 1 release from Figure 4.23.
4.25 Real Data: Chemical 1 Release. Using data from Figure 4.24, set threshold to eliminate 99% of false positives. Red is remaining plume, light blue is false positive. RSS retains the most true positive pixels.

4.26 Real Data: Chemical 6 release with ACE map. If we set a low ACE threshold, we can guarantee to detect all of the plume but also have a lot of false positives. A higher ACE threshold would miss some of the plume.

4.27 Real Data: Comparing all metrics for Chemical 6 release from Figure 4.26

4.28 Real Data: Chemical 6 Release. Using data from Figure 4.27, set threshold to eliminate 99% of false positives. Red is remaining plume, light blue is false positive. None of the metrics performed particularly well but it is difficult to tell, based on ACE, if plume is actually present or not in this image.

4.29 Embedded Data: Plots % of False Positives not overlapping in ACE scores vs RSS and SSE as a metric over 1050 different embedding examples.

4.30 Real vs. Embedded Radiances

4.31 Real vs. Embedded Radiances
List of Tables

3.1 SAM of Test Case Signatures ............................................. 38
3.2 Part of the Confusion Table for CSARIN_25T taken at 64 bands ... 57
3.3 Most Likely Predicted Table at Noise Power $1 \times 10^{-11}$, taken at 128 bands ................................................................. 58
Chapter 1

Introduction

The primary objective of this thesis is to evaluate chemical plume identification techniques through the study of their residuals. We are given a set of radiance data from a hyperspectral imaging device and we process it with a detection algorithm, which we will discuss later. This outputs a range of scores which we threshold in order to determine the location of a chemical plume. After thresholding, we study the residuals for false alarm mitigation, as well as determining if the chemical detected is in fact that specific chemical.

1.1 Hyperspectral Imaging

The typical human can only perceive radiance data within the visible spectrum, roughly 390nm to 750nm in wavelength (13,333cm\(^{-1}\) to 25641cm\(^{-1}\) in wavenumbers). In spectroscopy, the wavenumber is defined as

\[ \tilde{v} = \frac{1}{\lambda} \]  

(1.1)
where \( \tilde{v} \) is the wavenumber and \( \lambda \) is the wavelength of the radiation. This is also the same for the typical camera, it can only capture what is considered visible. Humans can very easily distinguish objects by size, shape and color by processing this visual information. The same cannot be said about computers, though. Based solely on data found in the visible spectrum, we would have to rely on the same information humans use (size, shape and color) in order to differentiate objects and to determine a threat but many times, this information alone is not enough. Using Hyperspectral Imagers, though, we can acquire radiance data from a much wider range and at a higher resolution. This added detail and information is crucial in the detection and identification of gaseous chemical plumes, many of which have distinguishing features within the Long Wave Infrared (LWIR) range, roughly between 900\( \text{cm}^{-1} \) to 1300\( \text{cm}^{-1} \) in wavenumbers. In Figure 1.1, we show various radiance data in the LWIR range for three different material types: sky, mountain and grass. We can see that mountain and grass are very similar but sky is a drastic departure from either of the other two radiance data points. This indicates the sky must have molecules which absorb radiance very strongly, such as water molecules.

![Figure 1.1: Some radiance data from a Hyperspectral Imager.](image-url)
1.2 Thesis Outline

In Chapter 2, we introduce the physical model of the radiance data. Using this model, we develop a mathematical model based on the three-layer radiance model. Using approximations which help simplify the model, we arrive at the form of the model from which we can develop detection, identification and false alarm mitigation algorithms. We also introduce several detection algorithms commonly used, including Mahalanobis distance, matched filters, the concept of spectral angles and the adaptive coherence/cosine estimator.

In Chapter 3, we analyze the Pacific Northwest National Laboratory Spectral Library to find the fundamental limits of detection. If the detection algorithms cannot differentiate chemical A from chemical B in the absence of noise, then it would be impossible to accurately detect the chemical A in an embedded or real world scenario. We attempt to find these confusers by clustering the library using the spectral angle (Section 2.5) as the distance metric. Confusers are defined as any chemical which a detector can potentially mistake as the target chemical. We experiment with K-means clustering (Section 3.3.2) as well as hierarchical tree-based clustering (Section 3.3.3). Neither method produced clusters in which every member chemical is a confuser to another member chemical. By modifying our clustering method, we were able to produce meaningful and useful clusters (Section 3.3.5). We also apply the same clustering techniques while varying the resolution (Section 3.4) and adding noise (Section 3.5). We find the minimum detectable amount by analyzing the transmission coefficient of the chemicals in Section 3.6.

In Chapter 4, we introduce least-squares unmixing of the gas from the background spectra (Section 4.1). We discuss assumptions about the residuals from estimation which should remain valid as long as the model is not violated (as in the case of a false
alarm). From there, we explore several tests and metrics to determine their usefulness in false alarm mitigation. These tests exploit additional information available in the model not yet explored in the matched filter or the adaptive coherence/cosine estimator, i.e., information found in the residuals after least-squares unmixing. By comparing all the metrics considered in Section 4.9, the significance of fit F-test, the sum of squared errors and the residual sum of squared errors (defined as $\text{RSS}_{\text{lof}}$ in the formal lack of fit F-test) prove to be most useful in real and embedded data. Typically, though, if a metric works well for real data, it does not work for embedded data and visa versa. Differences in false alarms from real and embedded data is explored in Section 4.10.

Finally, in Chapter 5, we summarize our results. Additional research ideas are included as a potential starting point for continued research and development in false alarm mitigation and identification algorithms.
Chapter 2

Background

2.1 Signal Models

Data used in this study for the detection and correct identification of gaseous chemicals come from Fourier Transform Infrared (FTIR) Hyperspectral spectrometers in the form of radiance data. We use the three-layer radiance transfer model to develop a mathematical representation of the radiance data captured by our sensors. We will also discuss some special cases of the radiance model which will simplify some of the algorithms used in practical applications.

2.1.1 Modeling the Radiance Data

The three-layer radiance model is prominently used to model the captured data, shown in Figure 2.1 [1]. We can view the three layers as the background, plume, and atmosphere between plume and the sensor. The first layer, background, includes everything behind a potential plume, such as trees, buildings, sky, grass, etc, up to the actual plume itself. The plume layer only contains the plume (or absent thereof,
if no plume is present). The last layer, atmosphere, contains the radiance due to the atmosphere itself. Every layer will add its own contribution to the radiance, as well as potentially attenuate the radiance from the layers behind it. We can view the radiance contribution from the background as radiance due to the background modulated by some atmosphere. The radiance from the plume is modeled as a black body, as described by the Planck function (shown again in (2.1) for reference). The last layer will take both the background and plume radiance, attenuate both by the atmosphere and also add on another radiance that is due to the atmosphere. The atmospheric radiance is modeled by another black body.

\[
B(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1} \tag{2.1}
\]

where \( h \) is the Planck constant, \( k \) is the Boltzmann constant and \( c \) is the speed of light.

One simplification we make is to assume each layer is at a single homogeneous temperature. This allows us to model the background, plume, and atmosphere as three distinct black bodies. The Planck function used to describe these black body radiances is

\[
B(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1} \tag{2.1}
\]
The black body modeled above is dependent on wavelength $\lambda$ as well as temperature $T$. Using this simplification, we can model the off-plume radiance as

$$L_{\text{off}}(\lambda) = L_a(\lambda) + \tau_a(\lambda)L_a(\lambda)$$  \hspace{1cm} (2.2)

where $L_a(\lambda)$ and $L_b(\lambda)$ is the radiance due to the atmosphere and background, respectively, and $\tau_a(\lambda)$ is the atmospheric transmittance. The transmittance of any substance is a physical property of the substance itself and must be measured. We assume the atmosphere is homogeneous, which allows us to model radiance as

$$L(\lambda) = [1 - \tau(\lambda)]B(\lambda,T)$$  \hspace{1cm} (2.3)

Equation (2.3) can be applied to any of the radiances discussed in this paper by using the appropriate temperature and transmittance.

When a plume is present, it will produce it’s own radiance, as well as attenuate the radiance of the background. As we’ve seen above, the background is attenuated by the atmosphere. Logically, we need an equation where the background is attenuated by both atmosphere and plume and the plume radiance needs to be attenuated by the atmosphere. This leads us to the following equation:

$$L_{\text{on}}(\lambda) = L_a(\lambda) + \tau_a(\lambda)L_p(\lambda) + \tau_p(\lambda)\tau_a(\lambda)L_b(\lambda)$$  \hspace{1cm} (2.4)

The equation for $L_{\text{on}}$ is very similar to that of $L_{\text{off}}$, except the modulated plume radiance has been included, as well as the plume’s effect on the background radiance.

To describe the transmittance of the plume, we utilize the Beer-Lambert Law, also known as Beer’s Law. This law defines the plume transmittance $\tau_p(\lambda)$, shown
here as reference.

$$\tau_p(\lambda) = \exp \left[ - \sum_{m=1}^{N_G} \gamma_m \alpha_m(\lambda) \right]$$  \hspace{1cm} (2.5)

where $N_G$ is the number of gas signatures, $\gamma$ is the concentration pathlength and $\alpha(\lambda)$ is the absorption coefficient. The concentration pathlength is the product of the plume density or concentration, $C_m$, as well as the plume depth as seen by the sensor, $\ell$. The absorption coefficient $\alpha(\lambda)$ is the wavelength-dependent signature unique to each gaseous species.

### 2.1.2 Useful Expressions of the Radiance Signal Model

The radiance signal model discussed so far in (2.2) and (2.4) are derived from the physical aspect of the problem. This radiance model is difficult to utilize in detection algorithms and practical applications, though, so additional, equivalent models have been developed. The most important ones in respect to the study have been described below.

**Plume Embedding Form**

There is a limited amount of actual chemical release data available. It would be unreasonable to have real data at every temperature, background, and chemical signature data point. In order to further test the false alarm mitigation metrics we will develop later in the study, we must be able to embed a plume into a background. In order to do this, we need to rewrite $L_{on}$ in terms of $L_{off}$, resulting in the following:

$$L_{on}(\lambda) = L_{off}(\lambda) + [1 - \tau_p(\lambda)] \left\{ B(\lambda, T_a) - L_{off}(\lambda) + \tau_a(\lambda) [B(\lambda, T_p) - B(\lambda, T_a)] \right\}$$  \hspace{1cm} (2.6)
The advantage of this equation is that we can use the off-plume radiance directly and the radiance of the atmosphere and background are not in the equation explicitly. We can also estimate the temperature of the atmosphere and embed the plume at the same temperature, which would allow us leeway as determining the atmospheric transmittance $\tau_a(\lambda)$ is a complex and difficult task not within the scope of this research.

**Signal Processing Form** The signal processing form of the radiance model highlights the difference between the on-plume and off-plume radiance. We can rewrite the radiance model as

$$L_{on}(\lambda) = \tau_a(\lambda)[1 - \tau_p(\lambda))] \left\{ B(\lambda, T_p) - L_b(\lambda) \right\} + L_{off}(\lambda) \quad (2.7)$$

Equation (2.7) also defines the limits of what is detectable. In particular, the first term of the equation must be non-zero. This requires that:

1. $\tau_a(\lambda) > 0$ (atmosphere isn’t opaque)
2. $\tau_p(\lambda) > 1$ (plume isn’t transparent)
3. $B(\lambda, T_p) - L_b(\lambda) \neq 0$ (Plume Temperature and Effective Background Temperature are not equal)

If these are not met, then the plume will be rendered invisible to detection algorithms, blending in perfectly with the background.
2.1.3 Special Cases for Practical Applications

The equations presented so far for on-plume and off-plume radiance models are difficult to use in most signal processing applications. The relationship between all the variables are non-linear and can be unreasonable to work with from a practical standpoint. In order to circumvent these issues, we consider some special cases to simplify the equations.

Linear Signal Exploitation Model

Here we will develop the radiance model that is used throughout the rest of the study.

1. Optically-thin plume approximation

Above, we’ve stated that the plume cannot be transparent in order to be detectable. On the other hand, though, many applications aim for maximum sensitivity when detecting very weak plume. We consider a plume to be weak or optically thin if

\[ \tau_p(\lambda) = \sum_{m=1}^{N_G} \gamma_m \alpha_m(\lambda) \ll 1 \text{ for each } \lambda \quad (2.8) \]

This allows a linearization of Beer’s Law (2.5) by using the first term from a Taylor expansion.

\[ 1 - \tau_p(\lambda) \approx \sum_{m=1}^{N_G} \gamma_m \alpha_m(\lambda) \quad (2.9) \]

The signal model (2.7) then becomes

\[ L_{on}(\lambda) = L_{off}(\lambda) + \sum_{m=1}^{N_G} \gamma_m \beta_m(\lambda) \quad (2.10) \]
where $\beta_m(\lambda)$, called the "in-scene target signature" for optically thin plumes, is defined as

$$\beta_m(\lambda) = \tau_a(\lambda) \left\{ B(\lambda, T_p) - L_b(\lambda) \right\} \alpha_m(\lambda) \quad (2.11)$$

Importantly, the model in (2.10) is now linear in the concentration pathlengths $\gamma_m$.

2. **Flat background emissivity approximation**

We can view the emissivity of the background, in general, as "flat" because it doesn’t usually have sharp spectral features, like most chemical signatures. We can express the radiance of the background as

$$L_b(\lambda) = \epsilon_b(\lambda) B(\lambda, T_b) \quad (2.12)$$

In the above equation, $\epsilon_b(\lambda)$ is the emissivity of the background and $T_b$ is the temperature. We can estimate the emissivity to be roughly 1. Using this approximation, (2.11) then becomes

$$\beta_m(\lambda) \approx \tau_a(\lambda) \left\{ B(\lambda, T_p) - B(\lambda, T_b) \right\} \alpha_m(\lambda) \quad (2.13)$$

This approximation is not valid for plumes composed of chemicals with larger molecules, though, because the spectra for these are broad and featureless [2], violating our earlier assumptions.

3. **Linear Planck function approximation** The thermal contrast term of the Planck function can be linearized when $|T_p - T_b|$ is small, in general less than
5°K. The dependence on wavelength is weak over the LWIR window, also, leading to the following approximation:

\[ B(\lambda, T_p) - B(\lambda, T_b) \approx C_B \nabla T \]  \hspace{1cm} (2.14)

where

\[ C_B = \frac{\partial B(\lambda, T)}{\partial T} \bigg|_{T=T_b, \lambda=\lambda_0} \]  \hspace{1cm} (2.15)

and \( \lambda_0 \) is a reference wavelength chosen to be in the middle of the band. With this last approximation, (2.11) becomes

\[ \beta_m(\lambda) \approx C_B \Delta T \tau_a(\lambda) \alpha_m(\lambda) \]  \hspace{1cm} (2.16)

If we include all three approximations, the signal model from (2.7) becomes

\[ L_{on}(\lambda) = \sum_{m=1}^{N_G} (C_B \Delta T \gamma_m) \tau_a(\lambda) \alpha_m(\lambda) + L_{off}(\lambda) \]  \hspace{1cm} (2.17)

This gives a "signal-plus-clutter" model which is used in most practical applications for chemical plume detection, identification, quantification algorithms.

### 2.2 Plume and Background Signal Models

Using the model developed above, we can now develop a sampled version of the model that is used for our algorithms. We begin with (2.17). We can denote \( \tilde{L}(\lambda) \) as the
radiance measured by the sensor. This radiance is defined as

\[
\bar{L}_{\text{on}}(\lambda) = \sum_{m=1}^{NG} (C_B \Delta T^\gamma_m) \{ [\tau_a(\lambda) \alpha_m(\lambda)] \ast p(\lambda) \} + \bar{L}_{\text{off}}(\lambda) + n(\lambda) \tag{2.18}
\]

where \( p(\lambda) \) is the spectral response function of the sensor and \( n(\lambda) \) is the noise measured by the sensor, such as sensor noise inherent in the system.

Now if we consider the case where the sensor measures radiance at \( n_b \) distinct wavelengths or bands centered at \( \lambda_k, k = 1, 2, \ldots, n_b \). We sample (2.18) at these bands and get

\[
x_k = \sum_{m=1}^{NG} g_k s_{mk} + v_k, \quad k = 1, 2, \ldots, n_b \tag{2.19}
\]

with the following definitions:

\[
x_k \triangleq \bar{L}_{\text{on}}(\lambda_k) \quad \text{(measured radiance)} \quad (2.20a)
\]

\[
g_k \triangleq C_B \Delta T^\gamma_m \quad \text{(chemical contract)} \quad (2.20b)
\]

\[
s_{mk} \triangleq \left[ [\tau_a(\lambda) \alpha(\lambda)] \ast p(\lambda) \right]_{\lambda=\lambda_k} \quad \text{(at-sensor gas signature)} \quad (2.20c)
\]

\[
v_k \triangleq \bar{L}_{\text{off}}(\lambda_k) + n(\lambda) \quad \text{(background clutter plus noise)} \quad (2.20d)
\]

We can rewrite this in a concise vector-matrix form as

\[
x = \sum_{m=1}^{NG} g_m s_m + v \triangleq Sg + v \tag{2.21}
\]

This model shows that the at-sensor radiance is a linear combination of the gas signatures superimposed on the background radiance plus some noise [3].
2.3 Whitening

As we’ve previously shown, the at-sensor radiance model is now given by

$$x = Sg + v$$

(2.22)

For the detection algorithm, in general we will need to de-mean and then whiten the data vector $x$ [4]. Whitening is required because $\text{Cov}(v) \neq \sigma^2 I$. In order to whiten the data, we define

$$\text{Cov}(x) = \sigma^2 C$$

(2.23)

where $\|C\| = 1$ or some other normalization in order to fix the value of $\sigma$. We then multiply (2.22) by $C^{-\frac{1}{2}}$ on the left, giving us

$$\tilde{x} = \tilde{S}g + \tilde{v}; \quad \tilde{v} \sim N(0, \sigma^2 I)$$

(2.24)

where $\tilde{x}$, $\tilde{S}$ and $\tilde{v}$ are the “whitened” version of the variables from Eq. (2.22). In practice, we do not know the quantities $C$ nor $\sigma$, so these need to be estimated from the off-plume radiance.

2.4 Mahalanobis Distance (MD)

The Mahalanobis Distance (MD) [1] is very similar to taking the Euclidean distance between two vectors, but we also take into consideration demeaning and the background covariance matrix $\Sigma_B^{-1}$. The MD of $x$ from its mean $\mu$ is

$$y_{MD}^2 = (x - \mu)^T \Sigma_B^{-1} (x - \mu)$$

(2.25)
By demeaning and multiplying by $\Sigma_B^{-1}$, we make this distance metric scale invariant. In practice, we do not know the mean $\mu$ and covariance matrix $\Sigma_B$ because we do not know the exact distribution of the data $x$. Instead, we use the maximum likelihood estimates, which are defined as

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} x_n \quad (2.26)$$

$$\hat{\Sigma}_B = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \hat{\mu})(x_n - \hat{\mu})^T \quad (2.27)$$

### 2.5 Spectral Angle (SA)

The spectral angle is defined as

$$SA = \cos^{-1} \left( \frac{s_i s_j^T}{\sqrt{(s_i s_i^T)(s_j s_j^T)}} \right) \quad (2.28)$$

where $s_i$ and $s_j$ are signatures. A simple way of viewing this is to visualize this as the angle between two vectors in $n$-dimensional space. We can also consider the Mahalanobis Cosine, which is defined as

$$SA = \cos^{-1} \left( \frac{s_i \Sigma_B^{-1} s_j^T}{\sqrt{(s_i \Sigma_B^{-1} s_i^T)(s_j \Sigma_B^{-1} s_j^T)}} \right) \quad (2.29)$$

where we take the background covariance $\Sigma_B$ into consideration, which allows us to more accurately calculate the angle between signatures after they’ve been embedded into a background.
The Spectral Angle Map (SAM) is a simple extension of the idea of the spectral angle. A SAM is a table where, at each index \((i,j)\) shows the spectral angle of signature \(s_i\) to signature \(s_j\). These maps are useful for visualizing clusters of signatures, which we will explore later in Section 3.3.

### 2.6 Adaptive Coherence/Cosine Estimator (ACE)

Throughout this study, we have used the Adaptive Coherence/Cosine Estimator (ACE) as the basis for detection. ACE measures the angle between two vectors (vectors created by the signature and the radiance spectra) and is defined as [1]

\[
y_{ACE} = \frac{((x - \mu)^T \hat{\Sigma}^{-1} s)^2}{(s^T \hat{\Sigma}^{-1} s)((x - \mu)^T \hat{\Sigma}^{-1} (x - \mu))}
\]  

(2.30)

where \(x\) is the radiance from the sensor, \(\mu\) is the mean of the pixel, \(\hat{\Sigma}\) the background covariance matrix and \(s\) the chemical plume signatures. We use this detector as our basis for false alarm mitigation using the other metrics described later in the paper.

A graphical interpretation of the ACE detector is shown in Figure 2.2. Here, we can see that, if the spectra under question falls within the angular threshold, it will be be labeled as a hit. If the spectra falls outside the angular threshold, it will be considered as background. We can see there are still opportunities for false alarms, though. If the magnitude of the spectra is small but the angle is within the threshold, it will still be labeled as a true positive.
2.7 Summary

In this chapter, we’ve covered the three-layer radiance model widely used in the hyperspectral community. It describes how radiance is modulated by the atmosphere as well as the plume, if present. Using this model, we applied simplifications and approximations in order to linearize the equations. This allows us to develop a mathematical framework in which we perform the rest of our analysis. This framework also allows us to develop the basic detectors we use, namely the Adaptive Coherence/Cosine Estimator, which is an angular based metric used to determine how strongly present a plume chemical at a specific pixel.
Chapter 3

Study of Chemical Spectral Library

3.1 Overview

The main goal of the study of the chemical library is to determine the fundamental limits of detectability and identification of a specific chemical given a spectral library. Let’s consider the case where the sensor sees a chemical signature in a completely noiseless environment. We want to determine to what level of accuracy our current sensors can detect and correctly identify the chemicals. In order to do this, we need to find which signatures are most alike and therefore, most likely to confuse the sensor if both are present. From this point on, we will call these other signatures "confusers" to our actual target signature. We attempt to find confuser signatures through clustering the signatures. We specify a spectral angle for the clusters in order to obtain differing levels of discernability between signatures of the chemical library. If two signatures are within the same cluster with angular threshold $\alpha$, then...
we should expect that a sensor which isn’t sensitive enough to detect an angular difference of $\alpha$ to be able to differentiate the two signatures, therefore reaching the sensor’s fundamental limit of differentiability. In order to create the clusters, we used several clustering methods, including K-means and hierarchical tree-based clustering. We attempt to cluster the pure form of the signature, as given by PNNL, downsampled to 68 bands. We also experiment with the resolution of the signatures, from 8 bands to 512 bands by powers of 2 between $900.2\text{cm}^{-1}$ to $1244.9\text{cm}^{-1}$. In Section 3.2, we give an overview of the PNNL Library and how it is created. In Section 3.3, we perform clustering using K-means clustering as well as hierarchical tree-based clustering. In Section 3.4, we discuss the effects of downsampling on the discriminability of the chemical signatures. In Section 3.5, noise is added to the downsampled signatures and clustering is performed again.

3.2 The PNNL Spectral Library

The chemical vapor signature library used for this study was created by Pacific Northwest National Laboratory, called the PNNL Library. [5] The library offers over 400 unique chemicals with sets of signatures at 5, 25, and 50°C with a wavenumber range of $6500\text{cm}^{-1}$ to $600\text{cm}^{-1}$. Each signature is composed of a statistical fit of multiple signature captures at varying concentrations. Signatures at 25°C are composed of 10 to 12 captures, while signatures at 5°C and 50°C are composed of 6 captures. The signatures are fitted to the Beer-Lambert law (3.1), also called Beer’s Law, to remove outliers, detect impurities and improve SNR [6].

$$\tau_p(\lambda) = \exp \left[ -\sum_{m=1}^{M} \gamma_m \alpha_m(\lambda) \right]$$ (3.1)
In this equation, $\tau$ is the spectral transmission function, $\alpha$ is absorption coefficient spectrum of the chemical, also known as the signature, which is wavelength dependent. The abundance of the chemical plume is $\gamma$, which is a combination of the concentration $C_L$ of the plume and the pathlength $l$, which represents the depth of the plume. This models a plume of $M$ gas elements. The signature captures are performed using Fourier transform infrared spectroscopy, or FTIR spectroscopy. The database was constructed with remote sensing in mind. Prior to this library, other vapor-phase chemical signature libraries lacked resolution, sample quantification or contained too much system noise or nonlinearities which limited the usefulness for quantitative remote sensing or analytical laboratory work.

To make the measurements to obtain the chemical spectral signatures, PNNL used a Bruker IFS 66v/S models. The measurements are taken using vacuum-bench spectrometers to minimize the spectral interference effects of H$_2$O and CO$_2$. PNNL also modifies the spectrometers in order to compensate for two error-producing artifacts inherent in the system. Both artifacts are related to the aperture of the system. The first artifact is from light which has already been modified by the interferometer but instead of leaving the system, it is reflected back towards the source. This light is then reflected off the metal aperture wheel and back through the interferometer and modulated again. This distorts the resulting signature at all wavelengths. The second artifact is from using a cooled mercury cadmium telluride (MCT) detector. Johns [7] and Giver et al. [8] noted that the detector sees the source as well as the aperture annulus that is near room temperature. This annulus acts as a blackbody radiator whose blackbody rays enter the interferometer off-axis and create a “tailing” effect at the lower frequencies. The PNNL have modified the spectrometers to correct for both of these artifacts.
The absorption intensity values of the signatures are corrected by creating Beer’s Law plots at each position along the x-axis (in this case, for every wavenumber sampled). The slope is then determined using a classical, linear, least-squares approach. At each position, the absorbance $A_i(\tilde{v})$ is plotted against the mixing ratio $C_i$ for the $i$ different concentrations. For example, BUTENE-1 is sampled at 1.01322, 8.5783, 15.65, 4.2529, 32.8 and 54.1 Torr (units of pressure) at a path length of 19.94cm, given by

$$A_i(\tilde{v}) = \varepsilon(\tilde{v})dC_i$$

This creates a linear plot to determine the absorption coefficient $\varepsilon(\tilde{v})$ from the fitted slope. PNNL used a data-weighing scheme to minimize the effects of the known nonlinearities of the system. Each measured data point, the absorption coefficient, is weighted by $T_i^2(\tilde{v})$, where $A = -\log(T)$, and all data with absorption $A(\tilde{v}) \geq 1.6$ are assigned a weight of 0. By using this weighting scheme, linear behavior is achieved by using lower concentration measurements for strong absorption features. Good signal to noise comes from the higher concentration samples, where sharp peaks receive a lower weight or are zeroed out altogether.

In their pure form, as given by the PNNL library, most signatures are orthogonal to each other, or display a large angle of discrepancy between signatures. This is shown in Figure 3.1, below. This shows a histogram of the spectral angle map (SAM), which is a table of all the angles between signatures. From the figure, we can see that most inter-signature angles lie within the $60^\circ - 90^\circ$ range. Very few signatures are within $20^\circ$ of each other.

In Figures 3.2, 3.3, and 3.4 we show some example signatures of three different chemical gases. These chemicals are Acetoin_25T, HCFC142B_25T, and EDB_25T, respectively. All three signatures shown are downsampled to 68 bands ranging from
Figure 3.1: Histogram of the SAM of all signatures in their pure form. From the exponential rise of the curve, we can see that most signatures have a greater than 50° angle between them. Graph taken at 68 bands.

900 to 1250 wavenumbers (cm\(^{-1}\)) by steps of 5.15 wavenumbers. All signatures provided by the PNNL Library are indexed using wavenumbers (cm\(^{-1}\)).
Figure 3.2: Plot of Acetoin_25T. An example of a chemical signature with 1 major spike and several smaller features.

Figure 3.3: Plot of HCFC142B_25T. An example of a chemical signature whose features are changing rapidly, causing sharp spikes even after downsampling.
Figure 3.4: Plot of EDB_25T. An example of a signature with only one major feature.
### 3.2.1 Downsampling

The signatures provided by the PNNL library are of very high resolution. To process the amount of data given by the library would be a task unreasonable for the extent of this study or for any real-time detection algorithm. To circumvent this issue, we downsample all the signatures in the library to the resolution given in an experimental release using Sensor A. The downsampling process is shown in Figure 3.5. In this example, we downsample BUTENE-1 to the 68 bands given by Sensor A. First, we show the entire signature as given by the PNNL spectral library. This signature ranges from 525.0252 wavenumbers to $6.5 \times 10^3$ wavenumbers (1.534 to 19.048 microns in wavelengths). We then window this signature to the spectral range we’re interested in. This spectral range depends on the experimental data given. For example, a data cube from Sensor A has only 68 bands ranging from 900 wavenumbers to 1250 wavenumbers (8 microns to 11 microns in wavelengths). This range is shown in the second plot of Figure 3.5. The third plot shows the Gaussian filter. To create this, a Gaussian curve is created centered to every point in Sensor A’s resolution, resulting in 68 Gaussian curves. The full width half max (FWHM) of the Gaussian filter is determined by the Sensor A data as well and calculated as

$$\text{FWHM} = \frac{1}{N} \sum_{i}^{N} \gamma_i - \gamma_{i-1} \quad (3.3)$$

for downsampling, where $N$ is the number of bands (68 in the example) and $\gamma$ are the wavenumbers at which the data is given in. The windowed signature is convolved with the Gaussian filter to create the downsampled signature shown in the bottom of Figure 3.5. We can see the sharp spectral features of the original high resolution are smoothed out in the low resolution signature. The overall shape of the signature
remains and major features are retained, though. This effect of downsampling is more
detrimental to signatures with sharp spectral features than those who are smoother
overall. An example of a smoother signature is shown in Figure 3.6. Some of the
sharp features of the signature are lost from applying the Gaussian filter but because
the overall signature isn’t very sharp to begin with, the downsampled signature is
still very similar to the original.

Figure 3.5: Conversion of BUTENE-1 from high-resolution to downsampled signature
used for this study. Downsamled to 68 bands.
Figure 3.6: Plot of the high resolution and low resolution downsampled signatures for Acetoin_25T.
3.3 Clustering

3.3.1 Overview

If the library forms clusters naturally, we can utilize these clusters in our detection algorithms in a 2-pass method. We first determine which cluster the unknown data belongs in using a lower resolution detector. Knowing the cluster it belongs to can possibly lead to some general information about the unknown gas signature, for example, if a particular cluster exhibits similar characteristics. Then using a higher resolution detector, we can pinpoint the exact signature. By using this two-pass method, we avoid running the detection algorithm for every signature in the PNNL Library.

In order to create the clusters, we try using several methods. First, we used K-means clustering but this did not produce consistent results due to the inherent randomness of the algorithm. We used a hierarchical tree-based clustering method, which did produce consistent results but some of the results did not fit the question we were trying to answer. Finally, we developed a new scheme to cluster the data, which is based on K-means clustering. We performed all clustering on signatures at 25°C unless a signature was not provided at that temperature, in which case we would use any one of the signatures provided.

3.3.2 K-Means Clustering

The general idea behind K-means clustering is as follows: choose K centroids, find the data points which are within a certain ”distance” using the specified metric and form those into a cluster. Then recalculate the centroids as the center of these clusters, recalculate the clusters until there is no change in the clusters. This technique is
useful and accurate if there are a known number of clusters, for example in clustering a background into its major parts: grass, foliage, road, buildings, etc.

The issue with clustering data we can’t visualize is that first of all, we don’t know if the data forms clusters naturally. There might not be any clusters at all. We also don’t know how many clusters to initialize the algorithm with. For this study, we wanted to create clusters all within a specific distance of each other. The distance metric we used is the spectral angle, which is used most commonly in our detection algorithms.

In order to circumvent these problems, we developed a K-means clustering algorithm which attempts to create K clusters using random signatures as their initial centroids. To create clusters, we simply place each signature in the cluster with the smallest distance from the centroid. We recalculate the centroids and repeat the clustering until the clusters no longer change. We find the maximum intra-cluster angle and adjust K appropriately, either increase or decrease. We repeat the algorithm until K no longer changes. We utilized a binary search method to determine the values of K.

Because the K-means clustering algorithm is inherently random due to the random initial centroids, we performed multiple clustering runs at each threshold and created probability tables of how likely a signature will be clustered with any other signature. For example, at a spectral angle threshold of 22°, there is at least a 95% chance DIOFORM_25T is clustered with T12DCE_25T. At this threshold, only 4 signatures could be mistaken for another signature with 95% probability. At 30°, this number jumps up to 29 signatures. These probabilities are created by running the K-means algorithm described above 100 times and finding the probability of being within the same cluster. The plots in Figure 3.7 show that, although on a magnitude scale, these
two signatures within a cluster with a very small angular threshold differ but based on angle, these two signatures are almost identical. The actual angle between these two signatures is $1.88^\circ$.

![Graph showing signatures for DIOFORM and T12DCE](image)

Figure 3.7: Signatures for DIOFORM, T and T12DCE. Based on the spectral angle, these two signatures are very similar, with an angle of only $1.88^\circ$, even though they have different magnitudes.

### 3.3.3 Hierarchical Clustering by Trees

The idea behind hierarchical clustering by trees is as follows: given a distance metric, find the 2 data points which have the shortest distance. Join these two data points, update the overall distance of this new data point (composed of 2 actual data points) with every other data point based on a specified linkage method. Find the next 2 closest data points and repeat the process until all data points have formed into a
hierarchical tree. In order to form the clusters, we can cut the tree at a height which corresponds to the distance desired.

There are several different linkage methods, of which we used single, average, and complete linkage. Single linkage updates the distance using the shortest distance. In other words, it looks at all the data points already in the cluster and will add to it the signature which is closest to the edge of the cluster. This produces very elongated results, which can potentially display very little similarity between signatures. Average linkage will take the average of all data points within a cluster and treat this as a centroid, much like in K-means clustering. It then updates the distance table using this averaged data point. Complete linkage updates the distance table based on the furthest distance. This is the opposite of how single linkage updates the distance table. Given 2 data points in a cluster, the distance to a third, unclustered point will be the furthest distance. Complete linkage will produce clusters with maximum similarities. Both the average and complete linkage methods produced reasonable dendrogram plots for the library data. Examples of clusters formed for CSARIN_25T are shown in Figure 3.8 for complete linkage and Figure 3.9 for average. We can see that using the complete linkage method offers more discernability in the clusters versus using the average linkage method. From this point on, all tree-based clustering results will use the complete linkage method in order to produce clusters with the most similarities. Figures 3.10, 3.11 and 3.12 show the dendrograms for the various linkage methods discussed above. In Figures 3.13, 3.14, 3.15 and 3.16, we can see the plots of the spectral angle map as an image of the unclustered library, clustering using the single linkage method, average linkage method and the complete linkage method. All clustering is performed at 40° for these plots. The indices along the left and bottom indicate different signatures and the indices change based on clustering.
We expect clusters to be formed along the diagonal of these plots. For example, in the average linkage method plot, we see several light blue boxes along the diagonal. These boxes show where the clusters are, as well as the size of each cluster. From the average linkage spectral angle map image in Figure 3.15, we can see the largest cluster at around index 250. The yellow blocks along this cluster, indexed at 40, 75, 130, 300 and 350 indicate these clusters are similar to this large cluster indexed at 250. Everywhere else along this cluster is red, indicating the other clusters are nearly orthogonal to the largest cluster. In Figure 3.17, we can see that as the angular threshold increases, the number of clusters drops dramatically between 20° and 100°.

![Figure 3.8: Cluster formed by Hierarchical tree-based methods with Complete linkage and a spectral angle threshold of 35°.](image-url)
Figure 3.9: Cluster formed by Hierarchical tree-based methods with Average linkage and a spectral angle threshold of 35°.

Figure 3.10: Hierarchical Tree using the Single Linkage method
Figure 3.11: Hierarchical Tree using the Average Linkage method

Figure 3.12: Hierarchical Tree using the Complete Linkage method
Figure 3.13: SAM of unclustered library

Figure 3.14: SAM of clustered library using Single linkage method
Figure 3.15: SAM of clustered library using Average linkage method

Figure 3.16: SAM of clustered library using Complete linkage method
Figure 3.17: Plot of Angular Threshold vs. Number of Clusters
### 3.3.4 Concerns about Clustering Methods

Although the clustering methods so far produced reasonable and somewhat useful results, it doesn’t answer the question we are most concerned with. That is, given a specific threat signature, how likely is it that another signature can be mistaken for it? The detection algorithms mainly use the spectral angle, therefore we need to find all the signatures within a certain angular range of the threat signature. Although hierarchical tree-based clustering gave us consistent results, it does not answer this question.

Consider this test case comprised of a small subset of the library. In Table 3.1 we show the spectral angle map between 8 signatures. The lower triangle of the map is just a reflection of the upper triangle. From this table, we can observe some oddities based on the objective of the study. Let’s say we create a cluster with an angular threshold of 50° to MESITYL_{25T}. We can see that every signature within this subset would be included in this cluster; all signatures are within 50° of MESITYL_{25T}. Then we should be able to assume all signatures within this cluster are within 50° of each other, but clearly that is not the case. For example, the spectral angle between

<table>
<thead>
<tr>
<th></th>
<th>MESITYL_{25°}</th>
<th>CH3CN_{25°}</th>
<th>CH3CN_{3°}</th>
<th>ETO3P_{25°}</th>
<th>ET3TOL_{25°}</th>
<th>CH3SH_{25°}</th>
<th>NNDEA_{25°}</th>
<th>MXYLENE_{25°}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>25</td>
<td>25</td>
<td>29</td>
<td>44</td>
<td>35</td>
<td>49</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>0</td>
<td>1.9</td>
<td>27</td>
<td>40</td>
<td>35</td>
<td>50</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>0</td>
<td>26</td>
<td>41</td>
<td>36</td>
<td>51</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>50</td>
<td>44</td>
<td>52</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>25</td>
<td>41</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>39</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

*Table 3.1: SAM of Test Case Signatures*
NNDEA_25T and ETO3P_25T is 52°. In Figures 3.18, 3.19 and 3.20, we see the dendrograms created using this subset of signatures and using the single, average, and complete linkage methods, respectively. In the figures, the numbers along the bottom correspond to the signatures as ordered in the spectral angle map. The numbers along the vertical axis are the cutoff values. These values can be translated to angular threshold by using the following equation.

\[
cutoff \text{ value} = 1 - \cos(\theta)
\] (3.4)

where \( \theta \) is in radians. We can see that, when using any of the linkage methods, these signatures will be grouped together. In order to solve this problem, we decided to modify the K-means clustering algorithm to better fit our goals.

![Dendrogram of Test Case Signatures, Single Linkage](image_url)
3.3.5 Modified K-means Clustering

The modified K-means algorithm is as follows. We create a threat library and use these as our centroids. As opposed to K-means clustering, these centroids are now fixed. We then create a spectral angle map of the threat library and find the smallest angle between any two threat signatures. Half of this will be the angular threshold. We then create a spectral angle map of the threats library against the rest of the library. We cluster accordingly based on the angular threshold. Using this method, many of the signatures will not be clustered but we will be able to determine which signatures are most likely to be confused with a threat signature. The unclustered signatures are not regarded as potential threats or confusers so their existence does not pose any detriment to the accuracy of the detector. A block diagram of the modified K-means algorithm is shown in Figure 3.21.

To test this method, we created a threat library composed of CSARIN, LEWISITE,
MUSTHD, MUSTHN3, SARIN, SOMAN, TABUN and VX. We found the minimum angle between any two threat signature is 16.3323°. The minimum angle between any threat signature and the rest of the library is 20.0331°. Already, we can see that if the detector is capable of differentiating one threat from another, it won’t confuse any non-threat signature as a threat in the absence of noise. On the other hand, if the detector cannot differentiate any threat from another, an additional 341 signatures from the library can also be mistaken as a threat with a library that contains 388 signatures. As an example, CLCH2CN_25T or T13DCLP_25T can be mistaken for LEWISITE_33T (or visa versa) at 24°.

In Figure 3.22, we can see that the number of unclustered signatures drops dramatically between 40° and 80°. At around 100°, there is almost no discernibility. But at 40°, there are still 293 signatures out of 380 which cannot be confused with any threat signature. In Figure 3.23, we create the histogram of the spectral angle map,
where we compare the angles between the threats and the non-threat signatures. We can see from this plot there are very few angles below the $40^\circ$ threshold. We can also see that the angles grow in a linear fashion, as opposed to an exponentially growth shown in the previous histogram shown in Figure 3.1. When using this method, the impact of increasing the angular threshold is less severe than before.
Figure 3.22: Angular Threshold vs. Number of Signatures Unclustered using the Modified K-means Algorithm.

Figure 3.23: Histogram of the SAM created by comparing the threat library vs the non-threat library using the Modified K-means Algorithm.
3.4 Effects of Spectral Resolution on Clustering

3.4.1 Objectives

In the band study portion of the library study, we want to determine the effects of downsampling on the discernibility of signatures. The study is performed from downsampling a high resolution signature from the PNNL Library to either 8, 16, 32, 64, 128, 256 or 512 bands. We perform the downsampling by applying a Gaussian Filter to the signatures. The clustering performed for these tests were hierarchical tree-based clustering.

3.4.2 Results

In Figure 3.24, we can see the histogram of the spectral angle map as the number of bands is increased from 8 bands to 512 bands by powers of 2. From these plots, we can see the angles between signatures increases as the number of bands increases. This also means that as more detail is added to the signature, the more orthogonal the signature is to all other signatures in the library. There is a limit at which this added detail doesn’t add significant additional discernibility between signatures. In Figure 3.25, we can see that at around 64 or 128 bands, the added detail starts to add very little in discernibility between signatures. This also shows that adding too many bands would be counterproductive. The added complexity would just add computation time to the detection algorithms for little benefit. There is a large jump in the number of clusters from 8 bands to 16 bands. This indicates that there is also a lower bound to the ideal number of bands.

In Figure 3.26, we show the CDF of the spectral angle map. From before, we know that having more than 100 bands doesn’t give much advantage and could impact
performance negatively so we only study bands less than 100. We see that, as the
number of bands grow, the number of signature-pairs with a small angle increases,
pushing the overall CDF towards the right. The lines indicate where half of the
signatures-pair angles lie. The corresponding vertical lines lie at $51^\circ$, $65^\circ$, $71^\circ$ and
$75^\circ$. This indicates that, at 5 bands, half of the angles are below $51^\circ$. Increasing
the resolution to 10 bands moves this angle up to $65^\circ$. We also see that increasing
the number of bands from 20 to 100 only gives us a 4° advantage in discriminability.

Figure 3.27 displays a similar plot but shows where 10% of the signatures-pair angles
lie. At 5 bands, 10% of the angles are below $22^\circ$. But at 10 bands, this angle jumps
up to $36^\circ$. Moving from 20 bands to 100 bands increases this angle from $43^\circ$ to $49^\circ$.

In Figure 3.28, we show an angular threshold plot. This displays how many
signatures are above a certain angular threshold at any given angle. From this plot,
we can determine the discriminability of the signatures at any given angular threshold.
For example, at $1^\circ$, all signature-pair angles are greater than this threshold. The graph
also shows that almost all signature-pair angles are below $120^\circ$. As a reference, there
are only 75466 signature-pair angles with 388 unique signatures. We also see there
is a limit as we saw before. There is little gain in terms of increasing spectral angle
when moving higher than 40 or 50 bands.
Figure 3.24: Histograms of SAM from 8 bands to 512 bands by powers of 2.
Figure 3.25: Number of Clusters vs Angular threshold for all bands studied

Figure 3.26: CDF of the spectral angle map taken at 5, 10, 20, etc. bands as indicated above. Lines indicate where 50% of the signature-pair angles lie. Vertical lines correspond to 51°, 65°, 71° and 75°.
Figure 3.27: CDF of the spectral angle map taken at 5, 10, 20, etc. bands as indicated above. Lines indicate where 10% of the signature-pair angles lie. Vertical lines correspond to $22^\circ$, $36^\circ$, $43^\circ$ and $48^\circ$.

Figure 3.28: Plot which shows how many signatures-pair angles are below a threshold. For example, all signature-pair angles are above $1^\circ$, while almost no signature-pair angles are above $120^\circ$. The bands at which the spectral angle are calculated are indicated in the legend.
3.5 Noise Study

3.5.1 Objectives

From the noise study, we want to observe the effects of adding noise to the library and see how it affects the ability to distinguish one signature from another. We add noise based on noise power. Adding noise based on noise power matches a real-world scenario, where you expect the noise from the sensor and environment to be at roughly a constant noise power. We also need to figure out a way to add noise realistically to the signatures; we assume that the signatures as given by the PNNL library are as noise-free as possible and the noise we observe in this study are due to the sensor. Another factor we now consider are the negative values of the signatures. Because the signatures are absorption coefficients, a negative value does not make physical sense. Therefore, we truncate all negative values of the absorption spectra to 0.

3.5.2 Adding Realistic Noise

In order to add sensor noise realistically, we have to convert the signatures into what the sensor is actually taking in from the scene. The hyperspectral sensors take in the radiance of the scene so we will need to add noise to this radiance. We first convert the signature to a transmittance by using Beer's Law (3.1) and multiply this transmittance by a black body as defined by Planck’s law set to 300K. We then add white Gaussian noise by a specific noise power to this radiance and reverse the math to solve for the absorption coefficient. The equation for this noisy absorption spectra is shown in (3.5), where $\alpha$ is the absorption spectra, $B(300, \lambda)$ is the Planck function evaluated at 300K, $\gamma$ is the concentration pathlength and $n$ is the white Gaussian noise we manually add.
\[ \alpha_n = \frac{\ln \left\{ e^{-\gamma n B(300, \lambda) + n} \right\}}{-\gamma} \]  

This noisy absorption spectra is then used to for the noise study. An example of this is shown below: Figure 3.29 shows a noise-free absorption signature for TEP\_25T. In Figure 3.30, we’ve added noise by following the method outlined above.

*Figure 3.29: Plot of TEP\_25T without noise.*
3.5.3 Adding Noise based on Noise Power

We first find the maximum and minimum signature powers. We define signature power as

$$\text{Power} = \frac{1}{n} \sum_{i}^{n} \alpha_i^2$$  \hspace{1cm} (3.6)

where a signature is defined as

$$\text{Signature} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$$  \hspace{1cm} (3.7)

and $n$ corresponds to the overall number of bands, where the bands are indexed $(1, 2, \ldots, n)$. The maximum signature power is 0.0429, while the minimum is $4.0450 \times 10^{-3}$.

Figure 3.30: Plot of TEP_25T with additive noise power at 1E-6.
This wide range of signature power is an issue while adding noise, as adding a small amount of noise compared to the maximum will quickly drown out most of the signatures. From the histograms in Figure 3.31, it is evident that most of the signatures have very low power, while only a few have power greater than $1 \times 10^{-6}$.

For the study, we start by finding the SAM and histogram of SAM for the entire library without any noise. This is shown in Figure 3.32. We then add noise by powers of 10, starting at $1 \times 10^{-18}$ up until $1 \times 10^{-1}$. The histogram and SAM of the signature library plus noise for the smallest amount of noise tested is shown in Figure 3.33. We add noise from a distribution which looks like

$$\text{Noise} = \sqrt{N_P}N(0, 1)$$

(3.8)

where $N(0, 1)$ is the Gaussian Distribution and $N_P$ is the noise power. At this noise
level, the histogram and SAM still look very similar, if not identical, to when there was no noise at all. But when we start adding noise power levels of $1 \times 10^{-9}$, shown in Figure 3.34, the SAM and histogram has changed significantly. The average intersignature angles are drifting towards $45^\circ$. At a noise power of $1 \times 10^{-5}$, shown in Figure 3.35, noise has completely overpowered the signatures and the histogram is completely centered at $45^\circ$. This makes physical sense as well: as more noise is introduced into a system, it becomes more difficult to differentiate one chemical from another. We also created a histogram of angles if the input was pure noise, set at noise power $= 1 \times 10^{-1}$. This is shown in Figure 3.36. The angles shown in this histogram are centered around $45^\circ$. We can see that the histogram of noisy signature angles has changed almost completely to the histogram of pure noise when the noise power is high enough.

*Figure 3.32: SAM and Histogram of entire library, noise power = 0*
Figure 3.33: SAM and Histogram of entire library, noise power = $1 \times 10^{-18}$

Figure 3.34: SAM and Histogram of entire library, noise power = $1 \times 10^{-9}$
Figure 3.35: SAM and Histogram of entire library, noise power = $1 \times 10^{-5}$

Figure 3.36: Histogram of the SAM of Pure Noise
3.5.4 Confusion Matrices

We created a series of confusion matrices based on adding noise by noise power. A confusion matrix is a matrix indexed with signatures along the horizontal and vertical axis, same as in spectral angle maps. Along the vertical axis is the actual signature and along the horizontal are possible confusers. The matrix is populated with the probability of confusing a signature along the vertical axis with any other signature, all shown along the horizontal. To create the table, we add noise to a signature and try to identify it, comparing to the uncorrupted signature database. If we set the noise power to zero, we expect the confusion matrix to be a diagonal matrix with 1’s along the diagonal and 0’s everywhere else. As we add noise, we expect the 0 values to increase and the 1’s to decrease, unless that particular signature is very difficult to be mistaken for another.

In an example, the confusion matrix for CSARIN is shown below in Table 3.2. The signatures are downsampled to 64 bands and noise is introduced into the signatures as described previously at a noise power of $1 \times 10^{-5}$. At this noise level, CSARIN is still most likely to be identified but it is not identified perfectly; a quarter of the input data will be misidentified as BENZOH. This is only part of the table; the entire table consists of 388 signatures but the rest of the signatures have zero probability of being misidentified and therefore excluded from the table.

An interesting result came up from these studies. At noise power levels greater than $1 \times 10^{-11}$, DODEC_25T is the most likely signature to be predicted, therefore it is most likely to be confused with any other signature when noise is added. A sample of one of these most likely predicted tables is shown below in Table 3.3. This table is created by summing vertically in the confusion matrix. The absolute maximum of this value is 388, that is, if every signature is mistaken to be the same signature. At
<table>
<thead>
<tr>
<th>Confuser</th>
<th>Probability of Confusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSARIN_25T</td>
<td>0.71</td>
</tr>
<tr>
<td>BENZOH_25T</td>
<td>0.2546</td>
</tr>
<tr>
<td>ALLYLOH_25T</td>
<td>0.0304</td>
</tr>
<tr>
<td>SARIN_25T</td>
<td>0.0042</td>
</tr>
<tr>
<td>PYRIDINE_25T</td>
<td>0.0006</td>
</tr>
<tr>
<td>ETHEXOH_25T</td>
<td>0.0002</td>
</tr>
</tbody>
</table>

Table 3.2: Part of the Confusion Table for CSARIN_25T taken at 64 bands

noise power equal to $1 \times 10^{-4}$ at 128 bands, DODEC_25T has a most likely predicted number of 306.1104, or about 78.89%. In Figure 3.37, we can see DODEC_25T looks noisy to begin with.

Figure 3.37: Plot of DODEC_25T, the most commonly predicted signature.
<table>
<thead>
<tr>
<th>Chemical</th>
<th>Likelihood of Prediction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DODEC_25T</td>
<td>16.6022</td>
</tr>
<tr>
<td>EDA_25T</td>
<td>3</td>
</tr>
<tr>
<td>DIMAMINE_25T</td>
<td>1.9748</td>
</tr>
<tr>
<td>CH3CN_25T</td>
<td>1.8024</td>
</tr>
<tr>
<td>PENT2OH_25T</td>
<td>1.2902</td>
</tr>
<tr>
<td>DIIFLUOR</td>
<td>1.2432</td>
</tr>
<tr>
<td>NO2_25T</td>
<td>1.122</td>
</tr>
<tr>
<td>TEXANOL_25T</td>
<td>1.1076</td>
</tr>
<tr>
<td>PRTHIOL1_25T</td>
<td>1.0942</td>
</tr>
<tr>
<td>SO3_25T</td>
<td>1.0742</td>
</tr>
<tr>
<td>THIOLANE_25T</td>
<td>1.0606</td>
</tr>
<tr>
<td>APINENE_25T</td>
<td>1.058</td>
</tr>
<tr>
<td>BPINENE_25T</td>
<td>1.0514</td>
</tr>
<tr>
<td>HCHO_25T</td>
<td>1.0424</td>
</tr>
<tr>
<td>MIC_25T</td>
<td>1.0324</td>
</tr>
<tr>
<td>PRTHIOL2_25T</td>
<td>1.0302</td>
</tr>
<tr>
<td>EEACET_25T</td>
<td>1.0274</td>
</tr>
<tr>
<td>MCF_25T</td>
<td>1.0252</td>
</tr>
<tr>
<td>NNDEANL_25T</td>
<td>1.023</td>
</tr>
<tr>
<td>AMPHET</td>
<td>1.0118</td>
</tr>
<tr>
<td>C13DCLP_25T</td>
<td>1.011</td>
</tr>
<tr>
<td>ISPROAM_25T</td>
<td>1.0104</td>
</tr>
<tr>
<td>PROIMINE_25T</td>
<td>1.0074</td>
</tr>
<tr>
<td>Diethanil_25T</td>
<td>1.005</td>
</tr>
<tr>
<td>NitroProp_25T</td>
<td>1.0044</td>
</tr>
<tr>
<td>DIPAE_25T</td>
<td>1.0042</td>
</tr>
<tr>
<td>PINOH_25T</td>
<td>1.004</td>
</tr>
<tr>
<td>ISOOCT_25T</td>
<td>1.0036</td>
</tr>
<tr>
<td>D2O_25T</td>
<td>1.003</td>
</tr>
<tr>
<td>ISOPACET_25T</td>
<td>1.0024</td>
</tr>
<tr>
<td>MUSTHN3_39T</td>
<td>1.0024</td>
</tr>
<tr>
<td>NBUTALD_25T</td>
<td>1.002</td>
</tr>
<tr>
<td>NEOHEX_25T</td>
<td>1.0016</td>
</tr>
<tr>
<td>EVE_25T</td>
<td>1.0014</td>
</tr>
<tr>
<td>MITCY_25T</td>
<td>1.0014</td>
</tr>
<tr>
<td>PROSPANAL_25T</td>
<td>1.0012</td>
</tr>
<tr>
<td>TFA_25T</td>
<td>1.0012</td>
</tr>
</tbody>
</table>

*Table 3.3: Most Likely Predicted Table at Noise Power $1 \times 10^{-11}$, taken at 128 bands*
3.6 Transmission Coefficient Study

So far, we’ve studied the PNNL hyperspectral library as given to us, as absorption coefficients. Hyperspectral imaging sensors, though, take in images as radiances. In order to obtain a more realistic idea of clustering possibilities as seen by the sensor, we must convert the absorption coefficients into transmission coefficients. This can easily be obtained by applying Beer’s Law (3.1), as we’ve previously done to add noise realistically. To do this, though, we have to make a choice of what concentration pathlength ($\gamma$) to use. In Figure 3.38, we can see that the choice of concentration pathlength will affect the angles seen between the transmission signatures. The CDF plots show that, as the concentration pathlength increases, the angles between the signatures decrease, which will make them harder to differentiate. The transmission coefficients used for this study are normalized transmission coefficients, as defined by

$$\hat{\tau} = 1 - \tau = 1 - e^{-\gamma\alpha}$$

(3.9)

3.6.1 Minimum Detectable Amount

For this part of the study, we are most concerned with the threat library consisting of CWAs. The minimum detectable amount is the quantity necessary in order to be identified with 99% accuracy. The term ”quantity” here is the concentration pathlength, which is a combination of the density of the gaseous plume and the depth of the plume as seen by the sensor. The minimum detectable quantity plots for the CWAs are shown below in Figure 3.39. To create these plots, we add noise at the specified noise power and attempted to identify the noisy input signature compared to the entire library. We start off with a small concentration pathlength
and gradually increase this number until it can be identified with 99% accuracy out of 1000 runs. To identify these noisy signatures, we compare the signature to the entire library and find the minimum spectral angle. The noisy CWA is identified as the signature which corresponds to this angle. In an actual detection/identification algorithm, though, a threshold will be used so this doesn’t correspond directly to those algorithms. The algorithm presented allows for a best-case scenario for identification. From Figure 3.39, we can see that as the noise power rises exponentially, the minimum detectable amount also increases exponentially. In other words, the more noise that is introduced at the sensor input, the more gas is required to be present in order for a correct identification, which makes physical sense.
Figure 3.39: Minimum Detectable Amount

The non-linearities of Beer’s Law will affect the result if the concentration pathlength is too high. This can be seen in Figure 3.40 and 3.41. If the concentration pathlength is kept low enough, such as at 10 ppm, then there are only a few signatures within the cluster. As the concentration pathlength increases, the cluster membership also increases but with an important distinction: there are no overlapping signatures between the two clusters except for the centroid signature, CSARIN. The clusters formed in the plots are formed from comparing angles between the transmission coefficients with an angular threshold of 40°. For comparison, the absorption coefficients are also shown in the figures.
Figure 3.40: Cluster of normalized Transmission Coefficients with concentration path-length of 10 ppm.

Figure 3.41: Cluster of normalized Transmission Coefficients with concentration path-length of 1110 ppm.


3.7 Orthogonality Study

In the previous study, we found that adding noise to the signature database will increase the orthogonality of the signatures. Increased orthogonality increases the differentiability of signatures, although because of the added noise, correct identification became difficult. We suspect the spectral angle is directly related to the amount of overlap between signatures. To verify this, we convert the signatures into a binary value of ”peaks” and ”valleys”, where peaks are defined at 10% of the maximum signature value. As an example, a plot of F134A_25T is shown in Figure 3.42. From this plot, we can see that the actual signature for F123A_25T is now converted into a binary signature shown in red. We use this binary signature to calculate the Hamming distance, where Hamming Distance is calculated as

\[ d_{st} = \left( \#(x_{sj} \neq x_{tj}) / n \right) \] (3.10)

where \# is a discrete binary summation. We use the Hamming distance to study how overlap affects the spectral angle between signatures. We expect that, as the Hamming distance increases, the amount of overlap to shrink and therefore, the angle to increase.

We first convert every signature in the library to a binary signature as described above. We find that, for signatures which have a single or only a few sharp spikes, the signature’s are orthogonal to mostly every other signature; the Hamming distance is not well correlated to the spectral angle. These signatures have a low percentage of 1’s compared to 0’s in their converted binary signature. An example of a signature like this is shown in Figure 3.43, which is a plot of CLFORM_25T. The sum of 1’s in this converted signature is only 5. If, on the other hand, there are many 1’s in the
Figure 3.42: A plot of F134A_25T showing the actual signature and the converted signature used for Hamming distance calculations.

converted signature, we can expect this signature to have many peaks, several wide peaks or the signature to look similar to noise overall. An example of a signature which displays this property is shown in Figure 3.44. In this plot of ET3TOL_25T, we can see that much of the signature is within 10% of the maximum and none of the signature actually goes down to zero, as it does in CLFORM_25T. The sum of 1’s in ET3TOL’s converted signature is 56 out of a maximum of 68.

After this conversion, we create a Hamming Distance Map, which is the same as a spectral angle map only we calculate the Hamming distance instead. At first, we expect these two maps to be well correlated but we discover that this is only the case for certain signatures. We attempt to correlate the Hamming distance vs. the spectral angle on a per signature basis and created a histogram of all the correlation coefficients, shown in Figure 3.45. This histogram displays a very wide range of correlation coefficient values, from -0.8 to 0.8. If the Hamming distance and the
Figure 3.43: Plot of CLFORM$_{25T}$, an example of a signature with only 1 sharp spike.

Figure 3.44: Plot of ET3TOL$_{25T}$, an example of a signature with several spikes.
spectral angle were well correlated, we’d expect sharp spikes at around the ends of this spectrum but this is not the case.

### 3.8 Summary

In this chapter, we’ve attempted to form clusters of similar chemical signatures based on their angular distance. We base our distance metric as an angle because our state of the art detector, the adaptive coherence/cosine estimator, measures the angular difference between the target and background spectra. We found that clusters of signatures do not form naturally. In other words, there are no clusters to be formed but instead, signatures are spread nearly evenly across space. We created a modified
K-means clustering algorithm in order to determine which chemicals can most eas-
ily be confused with a chemical warfare agent. We also tested how discriminability 
changed as a function of spectral resolution and if we added white Gaussian noise 
to the signatures. We converted the signatures from absorption coefficients to trans-
mission coefficients, the form of the signature used in the detection algorithms, and 
found the minimum detectable amount. Any concentration pathlength less than this 
will be rendered undetectable by our current algorithms.
Chapter 4

False Alarm Analysis

An important aspect of the detection algorithm is correctly identifying what has been detected. The detection of an abnormality in the radiance data can be easily accomplished by whitening the data and then measuring the Mahalanobis distance. But in order to assess the likelihood that what is detected is an actual chemical plume, or the correct chemical, is a difficult task to overcome with certainty. We begin the analysis with a least squares approximation of the pixels in question to the model we developed in Section 2.2 and will delve into various metrics to test whether the pixel in question is a true positive or false alarm.

False alarms can arise from a few different scenarios. In the first case, we can have a background pixel whose demeaned and whitened radiance appear similar to the chemical signature to be detected, causing false alarms. These similarities could arise due to the material composition of the background, a whitening scheme that failed to completely whiten the background or just due to noise. In the second case, the false alarm could be due to another chemical plume. For example, we’ve seen in Figure 3.9, there are at least three different signatures which are very similar to
CSARIN, two of which are other CWAs and one of which is harmless.

4.1 Least Squares Unmixing

We view each pixel as either a linear mixture of gas spectra and background clutter or just background itself. In order to unmix the gas spectra from the background spectra, we treat this as a linear estimation problem. Suppose we have already detected a gaseous chemical plume and made a preliminary guess as to what the chemical could be. Recall from (2.22) that we model the data as

\[ x \sim Sg + e \quad (4.1) \]

where \( e \) are the actual residuals. We need to estimate the abundance \( g \) based on the observation \( x \). We denote the estimate of \( g \) as \( \hat{g} \) and the estimate of \( x \) as \( \hat{x} = S\hat{g} \) with the error

\[ \hat{e} = x - \hat{x} = x - S\hat{g} \quad (4.2) \]

and \( S \) is the signature matrix.

For least squares estimation (LSE), the estimate \( \hat{x} \) is uniquely determined by the perpendicular drawn from the tip of the vector \( x \) to the hyperplane created by the columns of \( S \). This gives us

\[ S^T \hat{e} = S^T(x - \hat{x}) = 0 \quad (4.3) \]

If the signatures included in \( S \) are all independent, where the signatures are the columns of \( S \), then there is a unique solution for \( \hat{g} \) such that \( \hat{x} = S\hat{g} \). This brings
us to the normal equations

$$(S^T S) \hat{g} = S^T x$$  \hspace{1cm} (4.4)$$

The inverse of $(S^T S)$ exists in this case. We can define LSE of $\hat{x}$ as

$$\hat{x} = S \hat{g} = S(S^T S)^{-1} S^T x = P_S x$$  \hspace{1cm} (4.5)$$

where the projection matrix $P_S$ is defined as

$$P_S \triangleq S(S^T S)^{-1} S^T $$  \hspace{1cm} (4.6)$$

The outline of LSE above is only valid if cov$(x) = \sigma^2 I$. If this is not true, we will need to use Generalized Least Squares Estimation instead, where we consider cov$(x) = \sigma^2 C$, where $C$ is a known symmetric positive definite matrix, such as the covariance matrix of $x$. We perform a square root decomposition of $C$ given by

$$C = C^{\frac{1}{2}} C^{\frac{1}{2}} $$  \hspace{1cm} (4.7)$$

and then rearrange and multiply Eqn. 4.2 by $C^{-\frac{1}{2}}$

$$C^{-\frac{1}{2}} x \sim C^{-\frac{1}{2}} S \hat{g} + C^{-\frac{1}{2}} e $$  \hspace{1cm} (4.8)$$

to whiten $x$, $S$ and $e$, which we will now refer to as $\tilde{x}$, $\tilde{S}$ and $\tilde{e}$, $\tilde{e} \sim N(0, \sigma^2 I)$. We can now solve for $\hat{g}$ in the same method as we did before

$$(S^T C^{-1} S) \hat{g} = S^T C^{-1} x$$  \hspace{1cm} (4.9)$$
where we can now state
\[ \hat{g} = (S^T C^{-1} S)^{-1} S^T C^{-1} x \] (4.10)

Another way to view generalized least square estimation, then, is to perform LSE in the whitened space. We also note that, if we were to only consider a single chemical and we demean the data \( x \), the generalized least square estimation of \( \hat{g} \) is the same as the Matched Filter (MF), defined as
\[ y_{MF} = \frac{(x - \mu)^T \Sigma_B^{-1} s}{s^T \Sigma_B^{-1} s} \] (4.11)

### 4.2 Model Validation

Given the model described in Section 2.2, we can exploit various aspects of the model in order to perform detection and identification. Currently, we use MF and ACE to perform detection, as illustrated in Figure 4.1. From the figure, we can see that MF is a projection of the pixel \( x \) on to the signature \( s \), whereas ACE is a measurement of the angle between \( x \) and \( s \). False alarms can easily arise, then, if the projection of the pixel \( x \) is relatively large, but in a completely different direction, or if the angle measured is relatively small, even if the length of the pixel \( x \) is either much larger or smaller than the signature.

These algorithms only exploit a very limited amount of data available in the model, though. By exploiting more information in the model, and more specifically exploiting information found in the residuals, we can check the validity of the model itself. With LSE (and generalized LSE), we can make certain assumptions about the error (residuals) \( \hat{e} \) [9]:

1. The error \( \hat{e} \) has zero mean.
Figure 4.1: Diagram showing effects of whitening and outlining what ACE and MF is composed of. We can see that the target spectra is simply the background spectra shifted over. This shifting is possible because of the linearization approximations we’ve included in the model. After demeaning and whitening, we are in the space shown to the right of the figure. MF looks at the projections of the test spectra on to the signature spectra, whereas ACE looks at the angle between the two.

2. The error $\hat{e}$ has a constant variance $\sigma^2$.

3. The errors are uncorrelated.

4. The errors are normally distributed.

If any of these assumptions are found invalid, the model for that pixel comes into question. Using this data, then, we can create metrics for false alarm mitigation.

### 4.3 Studentized Residuals

We first consider the residual radiance from the whitened estimates of $x$:

$$ \tilde{z} = \tilde{x} - \tilde{S}\hat{g} \quad (4.12) $$
where we recall that $\tilde{x}$ and $\tilde{S}$ are the whitened versions of $x$ and $S$ from Section 2.3. Note that $\tilde{z}$ is different from $\hat{e}$, which is obtained if we have perfect knowledge of the covariance and mean of $x$. In other words, $\tilde{z}$ is the whitened version of the estimated residuals $\hat{e}$. As long as the model assumptions remain valid, the residuals are jointly normal random variables with zero mean and covariance

$$
\text{Cov}(\tilde{z}) = \sigma^2(I - P) \tag{4.13}
$$

where $P$ is defined as the whitened projection matrix

$$
P \triangleq \tilde{S}(\tilde{S}^T \tilde{S})^{-1} \tilde{S}^T \tag{4.14}
$$

The studentized residuals are then defined as

$$
\hat{r}_i = \frac{\hat{z}_i}{s \sqrt{1 - p_{ii}}} \tag{4.15}
$$

where $p_{ii}$ are the diagonal elements of the projection matrix defined in (4.14) and $s$ is defined as

$$
s^2 = \frac{\text{RSS}(\hat{g})}{p - 1} \tag{4.16}
$$

and

$$
\text{RSS}(\hat{g}) = \sum \|\tilde{x}(\lambda) - \tilde{S}(\lambda)\hat{g}\|^2 \tag{4.17}
$$

By studentizing the residuals, we expect the variance to be unity for pixels which fit the model and variances to be markedly larger than unity for pixels which do not fit the model, i.e. false positives. After studentizing, the residuals also have no relation with any other variable, such as wavenumber, gas amplitude, or fitted values. If there
were any evidence otherwise, we would consider the pixel to potentially be a false positive.

To start off the analysis, we visually inspect the studentized residuals to see how "white" they are. Any trends found within the residuals would indicate the model is at fault and potentially a false alarm. We can see from Figure 4.2 the studentized residuals for the plume are nearly white, but not perfectly white due to an imperfect (non-infinite sample) inverse covariance matrix. The studentized residuals for embedded confusor, on the other hand, are all highly correlated with each other. Unfortunately, due to studentizing, variances of the residuals for both plume and confusor are all roughly unity, as shown in Figure 4.3. Although the studentized residuals for the confusor all look nearly identical, when analyzed individually, the residuals look nearly as white as the studentized residuals for plume.
Studentized Residuals: True Positive

Studentized Residuals: False Positive

Figure 4.2: Embedded Data: Studentized Residuals for Chemical 1 and Chemical 2. Clearly, the studentized residuals for plume pixels are nearly white, whereas the residuals are highly correlated for the confuser.

Figure 4.3: Embedded Data: Variances of Studentized Residuals. Very difficult to separate due to studentizing, which forces the variance to unity.
4.4 Formal Lack of Fit F-Test

In a formal Lack of Fit (LoF) F test, we consider a group of pixels where the plume is presumably spread over. We assume the plume is slow moving and can view each pixel as a replicated experiment. Thus, we can apply the standard LoF test to it. In order to implement this, we consider a sliding window where we consider the plume amount should be constant. We then estimate a single plume amount $\hat{g}$ for all the pixels in the block by minimizing the residual sum of squares:

$$\hat{g} = \arg \min_g \text{RSS}(g)$$

$$\text{RSS}(g) = \sum_{i=1}^{N_{\text{pix}}} \| \tilde{x}_i - \tilde{S}g \|^2$$

where $N_{\text{pix}}$ is the number of pixels in the sliding window. We can split the residual sum of squares as

$$\text{RSS}(\hat{g}) = \text{RSS}_{\text{rep}} + \text{RSS}_{\text{lof}}(\hat{g})$$

where

$$\text{RSS}_{\text{rep}} = \sum_{i=1}^{N_{\text{pix}}} \| \tilde{x}_i - \langle \tilde{x} \rangle \|^2$$

$$\text{RSS}_{\text{lof}}(\hat{g}) = \sum_{i=1}^{N_{\text{pix}}} \| \langle \tilde{x} \rangle - \tilde{S}\hat{g} \|^2$$

where $\langle \tilde{x} \rangle$ is the mean radiance over all pixels within the sliding window. We expect $\text{RSS}_{\text{rep}}$ to be orthogonal to $\text{RSS}_{\text{lof}}$, which is why we can apply the F-test to this scenario. $\text{RSS}$ contains $pN_{\text{pix}} - 1$ degrees of freedom, $\text{RSS}_{\text{rep}}$ has $pN_{\text{pix}} - p$ degrees of
freedom and $\text{RSS}_{\text{lof}}$ has $p - 1$ degrees of freedom. This gives us

$$\frac{\text{RSS}_{\text{lof}}/(p - 1)}{\text{RSS}_{\text{rep}}/(pN_{\text{pix}} - p)} \sim F(p - 1, pN_{\text{pix}} - p)$$

(4.23)

### 4.5 Band-based Metrics

We attempted to create some band-based metrics to help mitigate false alarms. Given a signature, such as the one shown in Figure 4.4 and 4.5, we expect the some bands to be more influential to the ACE detection score than other bands. Specifically, bands which are peaks and valleys influence detection results more so than bands which are near 0. Knowing this, we created an influential bands index which highlights the importance of these bands. An example of the influential band index is shown in Figure 4.6, where the main peak and valley of the whitened signature is captured, along with some of the minor peaks and valleys. We can then do several things, focusing on these bands. We chose to sum studentized residuals over these bands, but we could also sum raw residuals or to check ACE scores if we only included these bands.
Figure 4.4: Unwhitened and Whitened signatures for Chemical 1 and Chemical 2, Confusor #2 based on Mahalanobis Cosine angles ($\angle = 32.0810^\circ$)

Figure 4.5: Unwhitened and Whitened signatures for Chemical 1 and Chemical 3, Confusor #5 based on Mahalanobis Cosine angles ($\angle = 47.7908^\circ$)
Figure 4.6: Whitened Chemical 1 signature with indication of where the influential bands are located.
4.5.1 Summing Studentized Residuals over Influential Bands

We would expect, by summing studentized residual over a pixel will give a reasonably small result due to a closer approximation. A larger sum would occur if the model doesn’t fit as well. Where the signature is close to 0, the residuals would look like white noise, giving rise to large values over these bands, confusing the results and causing separation to be difficult. This difference would be accentuated if we focus solely on the influential bands.

When we use the summation of studentized residual errors over influential bands as a metric, or $\text{RSS}_{st-\text{inf}}$, we can see that it does offer some level of separability in Figure 4.8 for this case where the whitened signature and confusor are very similar. Unfortunately, though, as shown in Figure 4.10, this metric doesn’t work very well for embedded data.
Figure 4.7: Embedded Background Image, Embedding Mask where red is Plume (Chemical 1) and light blue is Confusor (Chemical 2), ACE Detection Map for Embedded Chemical 1 and Chemical 2.

Figure 4.8: $RSS_{st-inf}$ vs ACE for Figure 4.7.
Figure 4.9: Embedded Background Image, Embedding Mask where red is Plume (Chemical 1) and light blue is Confusor (Chemical 3), ACE Detection Map for Embedded Chemical 1 and Chemical 3

Figure 4.10: $RSS_{st-inf}$ vs ACE for Figure 4.9
4.6 Significance of Fit (SoF) F-Test

A useful statistic to test the fit of the least squares estimate is by comparing the mean square due to regression to the mean square due to residual variation [10]. If we calculate the mean values of both of these terms, we get

\[ E(MS_{\text{reg}}) = \sigma^2 + g^2 \sum (\tilde{x} - \langle \tilde{x} \rangle)^2 \]  
\[ E(s^2) = \sigma^2 \]  

where \( MS_{\text{reg}} \) is the mean square due to regression, \( s \) is the sum of squared errors (SSE), \( E(Z) \) is the mean value of the random variable \( Z \) and \( g \) is from (2.22). We now suppose that the errors \( e \) are independent \( N(0, \sigma^2) \) random variables. We then know that, if \( g = 0 \) (here, \( g \) is scalar because we only consider a single gas scenario), \( MS_{\text{reg}} \) multiplied by its degrees of freedom and divided by \( \sigma^2 \) follows the \( \chi^2 \) distribution with the same number of degrees of freedom. In our case, the degrees of freedom is 1. We also know that \( (N_b - 2)s^2/\sigma^2 \) follows a \( \chi^2 \) distribution with \( N_b - 2 \) degrees of freedom, where \( N_b \) is the number of spectral bands. We can then divide the two, which should follow an \( F \) distribution

\[ F(1, N_b - 2) = \frac{MS_{\text{reg}}}{s^2} \]  

Therefore, we can use this metric to test if \( g = 0 \) or \( g \neq 0 \).
4.7 SSE

The sum of squared error (SSE) of a pixel, as discussed previously, is defined as

\[ SSE = \| \tilde{x} - \tilde{S} \hat{g} \|^2 \] (4.27)

To use this as a metric, we calculate the SSE for each individual pixel. Intuitively, we would expect a higher SSE for false alarms and lower SSE for true positives, as these should have a better fit. Another view of this is to look at the variance of the raw residuals. The variance for pixels fitting the model should be lower because the estimate follows \( \tilde{x} \) closer. We can see this in Figure 4.11, where we embed both Chemical 1 and Chemical 2 and calculate the variance of the raw residuals. The ACE Detection map for this is shown in Figure 4.7. In Figure 4.12, we can see that SSE and the variance of Raw Residuals gives exactly the same results.

![Figure 4.11: Embedded Data. Radiance vs Raw Residuals. Shows how Raw Residuals of false positives display more variation than true positives](image-url)
Figure 4.12: Embedded Data. ACE vs SSE and ACE vs Variance of Raw Residuals. We can see that the variance of raw residuals is a scaled version of SSE.
4.8 RSS

We can use the deviation of the estimate from the mean of \( \tilde{x} \) as another metric. Previously, this was defined as RSS_{lof} in the formal LoF test. We now consider this value on a per pixel basis as opposed to over a block of pixels. One thing to note about this metric is that we are comparing an estimate of the radiance data to the mean of a single pixel of radiance data, i.e. a single value. Empirically, we expect the RSS of false positives to be higher than the RSS values for true positives. An example of this is shown in Figure 4.13. From this point on, we will refer to this metric simply as RSS.

\[ \text{Figure 4.13: Embedded Chemical 1 and Chemical 2. ACE vs RSS. We can see RSS is generally larger for False Positives (Chemical 2) compared to True Positives (Chemical 1).} \]
4.9 Comparison of Metrics

Unfortunately, no one metric works for both real and embedded data simultaneously. If the metric works well for real data, then typically it doesn’t work well for embedded data and visa versa. This could be because real false alarms present in the datasets we have are composed of background pixels which have similar enough spectral features to the target chemical’s spectrum that ACE gives it a high score. False Positives from embedding, though, come from another chemical signature which is very similar based on the Mahalanobis cosine. This difference in the source of false alarms makes a huge difference in false alarm mitigation, as we will show.

In order to compare metrics, we have two datasets. The first (Dataset A) is a series of embedded images, where we embed a plume along with 5 confusors based on similarity from Mahalanobis cosine in 5 separate images. In other words, each image we embed a plume along with a single confusor. We embed in a checkerboard pattern either over the foreground (grass) or over the mountain range. This allows us to test the metric both with less (grass) and more (mountain) atmosphere between the sensor and “plume”. The checkerboard pattern also allows us to have very similar background spectra for both plume and confusor. We use ACE as our base metric, which means after embedding, we aim to have similar ACE scores. This is to make the case more realistic: if we cannot separate plume from confusor based on ACE, we must rely on another metric for false alarm mitigation. In order to do this, though, we have to embed the confusor at a higher abundance $\hat{g}$ than the plume. An example of embedded plume and confusor are shown in Figure 4.14. We will be analyzing embedded data first.

The second dataset (Dataset B) is a chemical release of Chemical 1 over time. This produced a series of 32 images. Because this is an actual chemical release,
we can not know exactly where the plume is present and where it is not. We can only approximate, based on ACE and other metrics, the location and strength of the plume. The only thing we’re certain of is the chemical that was released. As such, we first process the image with ACE detection set to detect this chemical. We then set a low threshold in order to capture everything that is plume, as well as some false positives. We manually select where we guess the plume is based on spatial characteristics and ACE scores. Anything above the ACE threshold outside of this selected area is considered a false alarm. We then run the various metrics to see if it’s possible to automate this separation.

First, we will look at embedded data for Chemical 1 vs Chemical 2 and Chemical 1 vs Chemical 3. From Figures 4.15 and 4.17, the Variance of Raw Residuals (or SSE) metric and the RSS metric provide the most separability. The \( \text{SoF} \) and \( \text{RSS}_{st-\inf} \) metrics, on the other hand, provide almost no separability in embedded data. In Figures 4.16 and 4.18, we can see how the detection mask would look like after thresholding. Because we know where the plume is exactly, we set the threshold to retain 99% of the plume. Above each detection map we show the percent of false positives left after thresholding. Based on this value, \( \text{SSE} \) outperforms all other metrics.
Figure 4.14: Embedded Chemical 1 (true positive) and Chemical 2 (confusor) vs the estimated radiance.
Figure 4.15: Embedded Data: Chemical 1 vs Chemical 2 (confusor #2). Comparison of Variance of Raw Residuals, SoF F-Test, RSS, and \( \text{RSS}_{st-inf} \) as metrics.
Figure 4.16: Embedded Data: Chemical 1 vs Chemical (confuser #2). Detection Masks after thresholding. Threshold set to capture 99% of plume pixels.
Figure 4.17: Embedded Data: Chemical 1 vs Chemical 3 (confusor #5). Comparison of Variance of Raw Residuals, SoF F-Test, RSS, and $RSS_{st-inf}$ as metrics.
Figure 4.18: Embedded Data: Chemical 1 vs Chemical 3 (confusor #5). Detection Masks after thresholding. Threshold set to capture 99% of plume pixels.
Another example is embedding Chemical 4 versus Chemical 5. This also shows that for embedded data, the SSE and RSS metrics works very well, whereas the SoF and RSS_{st-inf} metrics do not provide any useful information. In Figure 4.19, we show the two signatures before and after whitening. These two signatures are overlapping in many parts with an angle of only 29.1294°. Figure 4.20 shows the embedding mask as well as the ACE detection map. This map shows a clear overlap in ACE scores, making differentiation impossible. Figures 4.21 and 4.22 show how the metrics compare to ACE and also the resulting detection maps. Again, SSE performs the best out of all metrics tested for embedded data.

![Figure 4.19: Unwhitened and Whitened signatures for Chemical 4 and Chemical 5, Chemical 4’s closest confusor based on Mahalanobis Angle (∠ = 29.1294°)](image_url)
Figure 4.20: Embedded Data: Chemical 4 vs Chemical 5 (confusor #1). Shows the Embedded Image, ACE Detection Mask and Embedding Mask. In Embedding Mask, Red is Plume (Chemical 4) and light blue is Confusor (Chemical 5).
Figure 4.21: Embedded Data: Comparing all metrics for embedding Chemical 4 with Confusor #1 (Chemical 5) from Figure 4.20
Figure 4.22: Embedded Data: Using data from Figure 4.21, we generated masks showing where "plume" is after thresholding. SSE can remove all of the false alarms, whereas SoF couldn’t remove any of the false alarms. Thresholding based on detecting 99% of the plume pixels.
Now we will examine real data. Figures 4.23 show the image captured by the sensor, as well as the ACE detection map. Below the ACE map is a Hits Mask, where any pixel that scored higher than 0.1 from the ACE map is considered as possibly plume. In Figure 4.24, we see that the SSE test does not perform well, but the SoF and RSS metrics provide useful information and separability. This is clearer in Figure 4.25, where we can see where the plume is after thresholding.

![Real Chemical 1 Release, Radiance](image1)

![ACE Detection Map](image2)

![Hits Mask (ACE > 0.1)](image3)

*Figure 4.23: Real Data: Chemical 1 release with ACE map. If we set a low ACE threshold, we can guarantee to detect all of the plume but also have a lot of false positives. A higher ACE threshold would miss some of the plume.*
Figure 4.24: Real Data: Comparing all metrics for Chemical 1 release from Figure 4.23
Figure 4.25: Real Data: Chemical 1 Release. Using data from Figure 4.24, set threshold to eliminate 99% of false positives. Red is remaining plume, light blue is false positive. RSS retains the most true positive pixels.
Figures 4.26, 4.27, and 4.28 are similar plots as the ones shown for the real release of Chemical 1, except this is a release of Chemical 6. This image, though, is much more vague as to where the plume could be. The results are still the same, though, with $SoF$ and RSS providing the best separability.

![Real Chemical 1 Release, Radiance](image1)

![ACE Detection Map](image2)

![Hits Mask (ACE > 0.1)](image3)

*Figure 4.26: Real Data: Chemical 6 release with ACE map. If we set a low ACE threshold, we can guarantee to detect all of the plume but also have a lot of false positives. A higher ACE threshold would miss some of the plume.*
Figure 4.27: Real Data: Comparing all metrics for Chemical 6 release from Figure 4.26
SSE Detection Mask after Thresholding, %False Positive = 0.031384

F Detection Mask after Thresholding, %False Positive = 0.47932

RSS Detection Mask after Thresholding, %False Positive = 0.36377

RSS_{st-inf} Detection Mask after Thresholding, %False Positive = 0.012839

Figure 4.28: Real Data: Chemical 6 Release. Using data from Figure 4.27, set threshold to eliminate 99% of false positives. Red is remaining plume, light blue is false positive. None of the metrics performed particularly well but it is difficult to tell, based on ACE, if plume is actually present or not in this image.
Because both metrics $RSS$ and $SSE$ both perform reasonably well for embedded data, we wish to determine which one is better. We embed into the image 1050 times with 105 different signatures, each embedded into either foreground or mountain range and each with 5 closest confusors based on Mahalanobis angle. This produced the results shown in Figure 4.29. Because we’re embedding at such a large scale, we cannot visually inspect each embedding map. We then base the validity of our results on how much the ACE scores overlap. What’s plotted is the amount of false positives whose ACE scores do not overlap with the true positives. In order to be a good confusor, we expect the ACE scores to overlap, therefore we want to focus on the lower end of the x-axis. For the y-axis, we have percentage of false positives left after thresholding for either $RSS$ or $SSE$. Thresholding is based on retaining 99% of the true positives. From this plot, we can see that $SSE$ outperforms $RSS$ as a metric separating false positives from true positives in an embedded environment. Thresholding by $RSS$ will give more than 10% false positives in the final image in roughly 33.8% of cases whereas $SSE$ will only pass the 10% false positive threshold in roughly 10.6% of cases.
Figure 4.29: Embedded Data: Plots % of False Positives not overlapping in ACE scores vs RSS and SSE as a metric over 1050 different embedding examples.
4.10 Real Data vs. Embedded Data

In this study, we’ve seen that some metrics work better for real data than embedded data and visa versa. Although RSS works for both embedded and real data, it does not perform well enough to be considered universally applicable. These differences are most likely due to the nature of the false positives in each case. In the embedding case, we embed an a chemical plume into an image along with a confusor chemical. Both plume and confusor are both chemicals from the PNNL Library. On the other hand, in the case of real data, we have a real release of a chemical along with background pixels with enough spectral similarities to the released chemical that they achieve a relatively high ACE score, causing false positives.

In Figure 4.30, we can see some of the differences when we embed Chemical 1 versus when we have an actual release of Chemical 1. We can see that the demeaned radiances look nearly identical, but this radiance is also nearly identical to a typical background pixel’s radiance. After whitening, though, we can see some major changes. Both Real and Embedded Radiance look nearly white except between 1000cm$^{-1}$ and 1075cm$^{-1}$, where the major valley and peak reside. The background pixel, though, looks white throughout. The ACE scores indicated are the spectral angle between the Real and Embedded Chemical 1 pixels. Both Real and Embedded pixels scored an ACE score of 0.4 in the ACE detection map. We can see that, if we whiten with the same inverse covariance matrix, specifically the one from the Embedded Image where we know exactly where plume and background is, the spectra are more similar than if whitening with different covariance matrices, such as in the middle plot of Figure 4.30. Figure 4.31 shows 4 different radiances, as well as their LS estimate. This plot highlights the differences between real and embedded false positives. The embedded false positive’s peak matches the actual signature’s peak,
shown in green, very well, whereas the background-based false positives peak does not. One interesting thing to note, though, is that all pixels here gave a very similar ACE score, highlighting some of the deficiencies of the algorithm.
Figure 4.30: Real vs. Embedded Radiances
Figure 4.31: Real vs. Embedded Radiances
Chapter 5

Summary

There main goal of this thesis was to understand the fundamental limits of detectability given a chemical spectral library such as the PNNL Library. We experimented with clustering the library using the spectral angle as the distance metric to determine which chemicals are confusers to another chemical. By creating clusters at varying angular thresholds, we can determine how sensitive a hyperspectral spectrometer needs to be in order to accurately differentiate two chemicals in an ideal case. After we’ve compiled clusters of confusers for specific chemicals, we then focused on false alarm mitigation. We embedded both the chemical plume as well as it’s confusers in the same image in order to achieve false alarms. We also have real data with real, background-based false alarms. We used both sets of data, real and embedded, to test the the false alarm mitigation algorithms.

In Chapter 2 we introduced the basics of radiance models and detection algorithms. We transformed the physical model of the radiance as seen by the sensors into mathematical equations we can use for our detection algorithms. We then develop the basis for the rest of the study, the Adaptive Coherence/Cosine Estimator.
In Chapter 3, we analyze the PNNL Library of chemical signatures. Clusters do not form naturally and most chemical signatures form a spectral angle of greater than 60°. Clustering using the K-means algorithm gave reasonable clusters but were inherently random due to the random choice of initial centroids. Clusters formed using Hierarchical Tree-based methods gave consistent results but the clusters formed did not give us the results needed, as shown in Section 3.3.4. In Section 3.3.5, we developed a modified K-means algorithm which fit our purposes. We found that, in their pure form, if we were to only consider the CWAs provided, we would only need a spectral angle sensitivity of 16.3323° to separate one CWA from another and 20.0331° to distinguish and CWA from a non-CWA.

In Chapter 4, we utilized knowledge of the library in order to find the most likely confusors for each real or embedded plume and attempted to distinguish the plume from the confusor. When using ACE as the sole detector, it can be very difficult to differentiate a plume from it’s confusors, as they receive identical ACE scores. We used Least Squares Estimation to determine an abundance for each pixel and then studied the residuals. We studied studentized residuals, band-based metrics, the SoF F-test, the $SSE$ and the $RSS$ as metrics for False Alarm Mitigation. We found that $SSE$ worked very well for embedded data but did not work for real data. For real data, the SoF F-test and $RSS$ worked nearly the same, although it’s difficult to say which is truly better as the exact location of plume is unknown. These difference arise due to the nature of the false positive in each case. In real data, a false positive came from a background pixel whose radiance looks similar enough to the chemical in question that it confused the ACE detector. In embedded data, the false positives were chemicals embedded into the same scene. These false positives were distinctly
different from a background based false positive, as shown in Section 4.10.

Additional areas of research still include the effects of atmospheric transmission on radiance data and detection results. Atmospheric compensation and other methods of minimizing the distortion effects of the atmosphere will help further improve our detection and false alarm mitigation algorithms. Other methods for false alarm mitigation can also be explored, such as varying band-based metrics.
Bibliography


