AUTOMATED ISOTOPE IDENTIFICATION ALGORITHMS FOR
LOW-RESOLUTION GAMMA SPECTROMETERS

BY

JACOB BENJAMIN STINNETT

DISERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Nuclear, Plasma, and Radiological Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Assistant Professor Clair J. Sullivan, Chair
Professor James F. Stubbins
Professor David N. Ruzic
Assistant Professor Lav R. Varshney
Handheld radio-isotope identifiers (RIIDs) are widely deployed for nuclear security applications, but these detectors generally have poor isotope identification performance. Most of these deployed detectors use low-resolution NaI scintillators due to their low cost and good efficiency. Alternative detection hardware could be used to improve performance, but better detectors are generally cost-prohibitive for wide deployment in this mission space.

However, a trained spectroscopist can use these low-resolution detectors to make much more accurate identifications than the RIIDs produce. For this reason, it has been suggested that these RIIDs could be significantly improved by changing the onboard identification algorithms.

To this end, a peak-based Bayesian classifier has been developed to perform automated isotope identification. This algorithm was constructed to mimic the manual identification that a spectroscopist would perform. This approach can handle challenges such as detector calibration drift and unknown shielding scenarios, is capable of identifying mixed radiation sources, and is computationally inexpensive enough to be feasible for deployment on handheld RIID systems. A method for easily generating isotope libraries that are coupled to the detector and to the feature extraction algorithm is presented as well.

This method is demonstrated on a broad variety of gamma-ray spectra, ranging from small calibration sources (< 1 μCi) to Category I quantities of special nuclear material. This algorithm is also benchmarked against the ANSI N42.34-2006 Standard for Handheld Identifiers as a part of the Department of Homeland Security Algorithm Improvement Program.
ACKNOWLEDGMENTS

There are several people I’d like to thank for their help and support.

Dr. Neil Shafer-Ray got me started in research when the closest thing I had to relevant work experience was a summer spent shoveling out stables. Without the years of research with you, I don’t know that I would have made it to grad school.

I’d like to thank my adviser Dr. Clair Sullivan for everything over the past several years. I couldn’t have asked for a more interested and patient adviser. If I’ve ever written anything comprehensible, it’s because of your feedback. Almost every employment opportunity I’ve had was because you introduced me to someone or because you put a word in somewhere.

I would also like to thank Dr. Tomasz Kozlowski for his generous donation of computation time on the campus cluster, saving me months of waiting for results, not to mention the occasional beer or coffee.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 2 FUNDAMENTALS OF GAMMA-RAY DETECTION</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Gamma-Ray Detectors</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Interactions of Radiation with Matter</td>
<td>12</td>
</tr>
<tr>
<td>2.4 Other Radiation Sources</td>
<td>22</td>
</tr>
<tr>
<td>2.5 Summary</td>
<td>25</td>
</tr>
<tr>
<td>CHAPTER 3 EXISTING ISOTOPE IDENTIFICATION METHODS</td>
<td>27</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>27</td>
</tr>
<tr>
<td>3.2 Library Comparison Algorithms</td>
<td>28</td>
</tr>
<tr>
<td>3.3 ROI Algorithms</td>
<td>31</td>
</tr>
<tr>
<td>3.4 Decomposition and Deconvolution Methods</td>
<td>32</td>
</tr>
<tr>
<td>3.5 Template Matching Methods</td>
<td>35</td>
</tr>
<tr>
<td>3.6 Neural Network Approaches</td>
<td>36</td>
</tr>
<tr>
<td>3.7 Other Comments on Existing Algorithms</td>
<td>38</td>
</tr>
<tr>
<td>3.8 Proposed Identification Algorithm</td>
<td>39</td>
</tr>
<tr>
<td>CHAPTER 4 FEATURE EXTRACTION, LIBRARY GENERATION, AND SPECTRA SIMULA-</td>
<td>40</td>
</tr>
<tr>
<td>TION</td>
<td></td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>40</td>
</tr>
<tr>
<td>4.2 Feature Extraction</td>
<td>41</td>
</tr>
<tr>
<td>4.3 Isotope Library</td>
<td>47</td>
</tr>
<tr>
<td>4.4 Spectra Simulation</td>
<td>51</td>
</tr>
<tr>
<td>CHAPTER 5 STATISTICAL APPROACH TO ISOTOPE IDENTIFICATION</td>
<td>58</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>58</td>
</tr>
<tr>
<td>5.2 Bayesian Classifiers</td>
<td>58</td>
</tr>
<tr>
<td>5.3 Peak Pairing</td>
<td>61</td>
</tr>
</tbody>
</table>
5.4 Likelihood Models ........................................ 63
5.5 Algorithm Output ....................................... 80

CHAPTER 6 RESULTS AND BENCHMARKING ALGORITHM PERFORMANCE ........................................ 82
  6.1 ID Demonstrations ........................................ 82
  6.2 Preparing the Benchmarks .............................. 89
  6.3 Algorithm Benchmark Performance .................. 95
  6.4 Performance Discussion ............................... 97
  6.5 Limits of Detection ..................................... 99

CHAPTER 7 CONCLUSION ...................................... 102
  7.1 Conclusions and Future Work ......................... 102
  7.2 Improving the Feature Extraction .................... 102
  7.3 Improving the Model Library ......................... 103
  7.4 Improving the Bayesian Classifier ................... 104
  7.5 Improving the Spectra Simulation .................... 106
  7.6 Improving the Benchmarking and Optimizations .... 106
  7.7 Other Applications ...................................... 107

CHAPTER 8 REFERENCES ...................................... 109

APPENDIX A IDENTIFICATION CODE ......................... 117
  A.1 Isotope Identification Project/ ....................... 118
  A.2 BRAIN/ .............................................. 122
  A.3 Library/ .............................................. 130
## LIST OF FIGURES

2.1 A low-resolution Co-60 spectrum .......................................................... 5
2.2 A formal definition of energy resolution ................................................. 6
2.3 Comparison of detector resolutions ......................................................... 7
2.4 A low-resolution Ba-133 spectrum ........................................................... 7
2.5 Absolute efficiency curves for a 3”x3” NaI detector ................................. 8
2.6 Comparison of three different detector types ............................................ 9
2.7 The decay scheme of Co-60 ................................................................. 12
2.8 The Compton scattering process ............................................................. 13
2.9 An idealized Compton continuum .......................................................... 15
2.10 A low-resolution Cs-137 spectrum ......................................................... 16
2.11 A bremsstrahlung spectrum ................................................................. 17
2.12 A low-resolution Th-232 spectrum ......................................................... 18
2.13 A low-resolution spectrum of WGPu .................................................... 19
2.14 Total linear attenuation coefficient vs energy .......................................... 20
2.15 Low-resolution spectra of unshielded and shielded Eu-152 ....................... 21
2.16 Terrestrial background radiation map of the US ..................................... 22
2.17 Cosmic background radiation and elevation maps of the US .................... 24
2.18 Effect of background subtraction on gamma-ray spectra .......................... 25

3.1 Performance of some commercial isotope identifiers ............................... 28
3.2 An example of poor calibration due to gain shift ...................................... 29
3.3 The effect of counting statistics on measured spectra ............................... 30
3.4 Demo of an ROI algorithm ................................................................. 33
3.5 An example of ML-EM deconvolution on a simulated spectrum ............... 34
3.6 An example of a PCA algorithm ........................................................... 35
3.7 A representation of a two layer feed-forward neural network .................... 37
3.8 A representation of the operation of a single neuron in a neural network .... 37

4.1 The bior2.6 wavelet ............................................................................. 42
4.2 The wavelet transform of a simulated Gaussian signal ............................. 43
4.3 A low-resolution Co-60 spectrum and its wavelet transform .................... 46
4.4 The raw peak library of Eu-152 ........................................................... 48
4.5 A shielded spectrum of Eu-152 ............................................................. 49
4.6 The effect of overlapping Gaussian peaks .............................................. 49
LIST OF TABLES

2.1 Example detector technologies and their resolution and efficiencies. .................................................. 9

3.1 Comparison of selected major gamma-ray emissions of $^{177}$Lu and $^{239}$Pu. These isotopes have 66 and 157 total gamma-ray emissions respectively. .................................................. 31

6.1 Model Posteriors for the Am-241 Spectrum ................. 83
6.2 Isotope Posteriors for the Am-241 Spectrum ................. 83
6.3 Model Posteriors for the Co-60 Spectrum .................. 84
6.4 Isotope Posteriors for the Co-60 Spectrum .................. 85
6.5 Model posteriors for the $^{133}$Ba and $^{152}$Eu spectrum. .......... 86
6.6 Isotope posteriors for the $^{133}$Ba and $^{152}$Eu spectrum. .......... 86
6.7 Isotope posteriors for the shielded $^{152}$Eu spectrum ........... 87
6.8 Model Posteriors for the WGPu Spectrum .................. 88
6.9 Isotope Posteriors for the WGPu Spectrum .................. 89
6.10 Model Posteriors for the Shielded HEU .................... 90
6.11 Isotope Posteriors for the Shielded HEU .................... 90
6.12 AIP sources used for benchmarking ......................... 93
6.13 Example F-scores ............................................. 94
6.14 Detection Limits of Co-60 Spectra .......................... 101
CHAPTER 1
INTRODUCTION

Since the September 11th terrorist attacks, the United States has spent on the order of $1 trillion on homeland security and anti-terror efforts; this includes approximately $150 billion on either failed projects or equipment that does not work as well as intended [1]. These failed projects range from $3.5 billion on motion sensor and camera networks along the US-Mexico border [2] to $230 million on prototype tests for new radiation portal monitors [3]. This also includes many millions spent on handheld radioisotope identifiers.

Handheld radioisotope identifiers (RIIDs) are used to detect and automatically identify radioactive material in a wide variety of settings. From shipping ports to explosive ordnance disposal to border security checkpoints, these identifiers are relied upon for security. Generally, these detectors use a sodium iodide (NaI) spectrometer because of their relatively low cost and reasonable efficiency over a broad range of field conditions.

There is a need to improve the identification performance of these detectors, ideally without substantially increasing their costs. Published evaluations of these commercial detectors have demonstrated that their identifications are not generally accurate [4, 5, 6]. Unfortunately, more recent evaluations of these commercial detectors have not been published, as that information is generally considered sensitive information since it could potentially point to operational shortcomings.

It has been suggested [4] that the focus for performance improvements should be the isotope identification algorithms used by these detectors. While the spectra they take are low-resolution, a trained spectroscopist can generally make highly accurate identifications using these spectra, implying that the ID algorithms are potentially problematic [7, 8]. The U.S. Department of Energy (DOE) requires trained spectroscopists to be on call at all times to analyze unknown spectra and resolve alarms [7], further suggesting that these identification algorithms are out-performed by knowledgable humans.
There are several challenges for any automated identification algorithm. First, NaI spectrometers have relatively low energy resolution, making it impossible to resolve many of the photopeaks in a spectrum [5] and limiting both the performance of the identification algorithms and the methods for identification that can even be applied [9]. While other scintillator materials have much better energy resolutions, their increased costs and, in many cases, their decreased absolute detection efficiency make them infeasible for the same wide-scale deployment as NaI detectors.

Second, NaI spectrometers are susceptible to temperature changes [10, 11]. As the temperature fluctuates, the energy calibration will change and the centroid of the peaks will shift significantly. Some identification algorithms are unable to handle the error in peak centroid detection, leading to incorrect identifications.

Third, in most use cases for these detectors there is an unknown amount of background radiation from terrestrial naturally occurring radioactive material (NORM), from cosmic sources, and from some manufactured materials. This can obfuscate the spectrum and generally makes feature identification in a spectrum more difficult. Some methods for isotope identification rely on background subtraction (see Chapter 3), in which a background measurement is taken separately from the source measurement. This introduces further uncertainties in the process and is not always feasible for the implementations of these detectors.

Finally, an issue for all isotope identification methods is that the source activity and attenuating materials between the source and detector are generally unknown. Attenuating materials have multiple effects on spectra. They reduce the total number of gamma-rays that reach the detector, making counting statistics worse and photopeaks harder to resolve. Shielding materials also preferentially attenuate lower energy gamma-rays, often to the point of removing lower-energy photopeaks from a spectrum. Finally, since the shielding between the detector and the source is generally not the same as the shielding between the detector and sources of background radiation, intermediate materials will generally decrease the signal to background ratio, further compounding correct identification.

The poor performance of modern isotope identification algorithms prompts the development of new approaches, particularly ones more similar to the process a spectroscopist would use to make manual identifications. To this
end, a Bayesian classifier approach has been developed. This approach was chosen so that the identification process of a trained spectroscopist could be mimicked. Due to its modular nature, this Bayesian framework also allows for the inclusion of other sources of information, which could be used to easily improve identification performance.

This dissertation is divided into six chapters. Chapter 2 describes the physics and hardware of gamma-ray detection, as well as the physical problems that hinder accurate isotope identification. Chapter 3 provides an overview into existing isotope identification algorithms. Chapter 4 is the beginning of our approach to isotope identification, and it details the feature extraction and the isotope library generation projects. Chapter 5 develops the Bayesian classifier that actually performs the identification step, and Chapter 6 demonstrates its performance. Finally, Chapter 7 contains directions for future work and concluding remarks on this project.
CHAPTER 2
FUNDAMENTALS OF GAMMA-RAY DETECTION

2.1 Introduction

From a physical perspective, there are many challenges for isotope identification algorithms to overcome. These phenomena can be categorized as intrinsic to the detector, a consequence of the interaction of radiation with matter, or from other radiation sources. This chapter describes each of these categories respectively. Discussion will be limited exclusively to gamma-ray detection.

2.2 Gamma-Ray Detectors

There are many different hardware types of gamma-ray detectors, but this dissertation will focus on thallium-doped sodium iodide scintillators (generally referred to as NaI detectors), as the majority of portable gamma-ray identifiers use NaI scintillators for reasons to be discussed here. The principles of how these detectors function are described exhaustively in [10]. Only a brief description of detection technologies and the properties that directly affect isotope identification are described here.

A gamma-ray detector is used to measure a spectrum, as in Figure 2.1. A spectrum is a histogram of radiation counts binned by detector channel (which maps to an energy bin). The detectors used for this dissertation produce spectra binned into 1024 channels, with an effective energy range from approximately 40 keV to 3000 keV. The features of the spectrum are later described in Section 2.3.

The two most important characteristics of these detectors are the energy resolution and the efficiency.
Suppose a detector is exposed to a monoenergetic source of gamma-rays with energy $E_0$. Ignoring scattering effects and assuming the spectrum is observed long enough to minimize channel-to-channel statistical fluctuations, the response of a NaI detector is approximately Gaussian, as shown in Figure 2.2. Define the full-width at half maximum $FWHM(E_0)$ at centroid (mean) $E_0$ of the Gaussian curve (note that $FWHM = 2.355 \cdot \sigma$, where $\sigma$ is the standard deviation). Then, the energy resolution $R(E_0)$ is then defined as:

$$ R(E_0) = \frac{FWHM(E_0)}{E_0} $$

An artificial comparison of a low-resolution versus a high-resolution response is shown in Figure 2.3. The resolution is fundamentally important to the problem of isotope identification for several reasons. First, many isotopes produce gamma-rays that are close in energy; for a lower-resolution detector, these peaks overlap, which makes detection of many of the peaks difficult or impossible. An example spectrum of $^{133}\text{Ba}$ is shown in Figure 2.4; this isotope produces photopeaks at 276, 302, 356, and 383 keV. The significant overlap makes these peaks difficult to detect for even high-count spectra. Second, even if an isotope doesn’t create peaks close in energy, other features in the spectrum such as Compton edges or peaks from other isotopes will often overlap, again rendering true photopeaks unresolvable.

The second detector characteristic to consider is the detector efficiency.
When a gamma-ray enters the detector volume, there is a significant probability that it will continue to pass through the detector either without interacting or only depositing a part of its energy within the detector. The probability that the gamma-ray will deposit its full energy into the detector is defined to be the absolute detector efficiency. This efficiency is a nonlinear function of gamma-ray energy but is generally decreasing with increasing energy; an example efficiency curve is shown in Figure 2.5. The absolute detector efficiency is the product of the geometric efficiency and the (detector’s) intrinsic efficiency. The intrinsic efficiency is the probability that a gamma-ray that is incident on the detector will be recorded by the detector, while the geometric efficiency is the probability that a gamma-ray emitted by an isotropic source will reach the detector at all. This is equal to the solid angle subtended by the detector face divided by $4\pi$, or the area of the detector face divided by $4\pi r^2$, where $r$ is the distance to the detector.

In general, the efficiency actually depends on the position of the radiation source relative to the detection volume. For a cylindrical detection volume, the path length of a gamma-ray traversing the detector will depend on its trajectory. However, for typical handheld detector usage, it is assumed that the source is in front of the detector and suitably far away, meaning that any gamma-ray from the source will travel approximately parallel to the major
axis of the detection volume.

The detector efficiency must be incorporated into the modeling for any isotope identification algorithm, as higher-energy photopeaks will be reduced in area and may be harder to detect in low-count spectra. For the identification algorithm in this dissertation, this problem is solved in the library generation step, discussed in Chapter 4.

For isotope identification, a high resolution, high efficiency detector would clearly be best. A few common detector types are described in Table 2.1 [11, 14]; there are many other detection technologies that range from much higher resolution (e.g. microcalorimeter detectors, 0.02% resolution at 103
Figure 2.5: Absolute efficiency curves for a 3"x3" NaI detector. The solid line represents the computed curve, while the circles represent experimental data; open circles are the data points not corrected for cascade summing. The absolute efficiency is equal to the detector efficiency times the geometric efficiency. The sources were placed 5 cm (top curve) and 17 cm (bottom curve) from the detector. Reproduced from [12].

keV [15]) to much lower resolution (e.g. plastic scintillators). While high purity germanium (HPGe) detectors seem ideal, there are operational challenges (e.g. the detectors must be kept extremely cold, generally with liquid nitrogen) and economic limitations (an HPGe detector may cost an order of magnitude or more than an NaI detector) to their deployment. NaI detectors are so prevalent due to their resolution and efficiency at their price point [6].

2.2.1 Applications

Because of the physical and economical limitations of gamma-ray detectors, it is important to choose a detection technology suited for the application. These applications can be generally categorized as alarming/detection, identification, and mapping [16]. These detectors can also be classified by total
Figure 2.6: Comparison of three different detector technologies with a mixed plutonium and iodine source [13].

Table 2.1: Example detector technologies and their resolution and efficiencies.

<table>
<thead>
<tr>
<th>Detector</th>
<th>Resolution @ 662 keV</th>
<th>Relative Conversion Efficiency Compared to NaI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NaI(Tl)</td>
<td>6-7%</td>
<td>100%</td>
</tr>
<tr>
<td>CsI(Tl)</td>
<td>5%</td>
<td>85%</td>
</tr>
<tr>
<td>LaBR₃(Ce⁺⁺⁺)</td>
<td>2.9%</td>
<td>160%</td>
</tr>
<tr>
<td>HPGe</td>
<td>0.3%</td>
<td>210%</td>
</tr>
</tbody>
</table>

detector size: pocket-sized, portable (but larger than pocket-sized), fixed position, or vehicle mounted.

- Pocket: pocket-sized detectors have a variety of applications, from simple alarming for single detectors to radiation mapping with networks of these detectors [17]. Because of the size, battery requirements, and electronic constraints, a medium to high relative efficiency per volume is needed, so these detectors typically use NaI or CsI, if not a simple Geiger counter. These detectors do not typically perform on-board isotope identification.

- Portable: Portable detectors (above pocket-sized systems) range from hand-carryable detectors such as the IdentiFinder [18] to backpack systems like the Bruker Radiation Backpack Sentry [19]. These detectors have a wide variety of applications, from detecting nuclear material con-
tamination to explosive ordnance disposal. In the past 15 years, NaI detectors have widely been deployed [6] due primarily to their cost, efficiency, and thermal stability, but other detector types are commercially available and are becoming more economically feasible. For sensitive applications with a higher budget, a medium to high resolution detector could be used, such as HPGe.

- Fixed position: Fixed units are usually large plastic scintillators and are used for vehicle and cargo screening [20, 21], although NaI has been studied as well [22, 23]. For radiation portal monitors (RPMs), very large detector volumes are necessary to detect as many gammarays as possible, as in these scenarios any emitted radiation is going to have passed through a large amount of shielding (e.g. a cargo shipping container). These volume requirements make most detector types unsuitable for this application. Ideally, identification would be performed by these detectors, but physical limitations greatly hinder performance [24]. These detectors would typically be used to create an alarm whenever a non-NORM source is detected [25] and investigate further with other hardware [26, 27, 28].

- Vehicle-mounted: For source detection and radiation mapping, a few projects have explored mounting arrays of detectors to moving vehicles. For example, ARES (Airborne Radiological Enhanced-sensor System) at Lawrence Berkeley National Laboratory uses an array of 26 CsI detectors to localize radiation sources from a helicopter [29, 30]. These typically have some identification capabilities, but their primary purpose is to locate sources and identify contaminated areas. For identification, a detector with a smaller stand-off distance could be used for further investigations of detected sources.

There are other detection technologies and applications that do not fit within these categories, such as ultra-high resolution microcalorimeter detectors for SNM enrichment measurements [15, 31, 32].

While there are many applications for gamma-ray detectors, this dissertation is focused on isotope identification for handheld NaI detectors. These detectors are already widely deployed for many applications, and as discussed in Chapter 1 have much room for algorithm improvement. The methods pre-
sented have been developed and benchmarked for NaI but are applicable for other detection technologies.

2.2.2 Detector Calibrations

For most identification algorithms, it is necessary to use calibrated spectra. The energy calibration is the function that maps the detector channel to an energy bin. It is generally found by using several calibration sources (of which the peak energies are known and easily detected), finding the peak centroids, and applying some polynomial fit. For NaI detectors, the channel \( i \) is usually mapped to energy \( E_i \) by a second-order polynomial:

\[
E_i = a + b \cdot i + c \cdot i^2
\]  

Some detectors instead use only a linear calibration, although this is generally inaccurate for low and high energy regions due to the well-documented nonlinear response of NaI detectors [33]. In some cases, a higher-order calibration or a calibration with a nonlinear term can be used.

This calibration is simple, but unfortunately it is not static. Changes in ambient temperature or bias voltage will shift this calibration, so a detector that is calibrated in the morning may not still be calibrated accurately in the afternoon. Even with automated calibration, the result may be poor. Some detectors are shipped with a built-in calibration source, such as \(^{137}\text{Cs} \) [34]. This gives the detector a single peak to calibrate with, but this is not enough to fit an off-set linear or quadratic calibration. This could be improved upon by providing an \(^{241}\text{Am},^{137}\text{Cs}, \text{and}^{232}\text{Th} \) source (providing calibration points at 59.5 keV, 662 keV, and 2614 keV respectively), but this will increase the cost of these detection systems further.

These calibration issues often pose significant challenges to identification algorithms, particularly those that rely on very accurate peak energy information. This is discussed further in Chapter 3.
Figure 2.7: The decay scheme of $^{60}$Co [12]. There are several decay paths to reach the ground state; for example, if one of these atoms decays, there is a 99.88% chance it emits a 318 keV $\beta^-$ particle, followed by an 1173 keV and 1332 keV gamma-ray.

2.3 Interactions of Radiation with Matter

To develop the identification algorithm that is the subject of this dissertation, the fundamental interactions of radiation with matter must be somehow included in the modeling. In addition to the effects of detector resolution and efficiency, a spectrum is complicated by the various modes of interaction, as well as by other sources of radiation.

When a parent nuclide decays, the daughter radionuclide is generally left in a highly-excited state, which (often) subsequently relaxes by the emission of gamma-rays. For example, $^{60}$Co decays by beta emission to $^{60}$Ni, and relaxes by emitting gamma-rays with energies 1173 keV and 1332 keV. The decay scheme is shown in Figure 2.7 and a spectrum was shown earlier in Figure 2.1. When the many 1173 keV or 1332 keV gamma-rays are fully absorbed by the detection volume, a photopeak is formed, which are well-approximated by a Gaussian curve for NaI detectors for reasons discussed elsewhere in [10, 11].

The gamma-ray emissions of a radionuclide are not variable because these nuclei can only exist in certain well-defined energy states. Since there are a finite amount of allowable energy steps and branching ratios (the probability of a given energy emission during the relaxation of one nucleus), an isotope
can be identified by comparing the peak energies and peak areas (which are proportional to the branching ratios) against a table of radiation emissions, such as Brown and Firestone [35].

If detector resolution and efficiency were the only effects to consider, the spectrum in Figure 2.1 would be a perfect Gaussian mixture from the photopeaks, but instead other features are observed.

2.3.1 Continuum Effects

Beyond photopeaks, a spectrum generally contains a combination of Compton continua and, in some cases, a bremsstrahlung continuum.

2.3.2 Compton Continuum

Compton scattering is an interaction between a gamma-ray and an orbital electron, which could be in the detector volume or in the intermediate material between the radiation source and the detector. The gamma-ray is inelastically scattered from the electron, which changes its trajectory and reduces the gamma-ray’s energy. This process is shown in Figure 2.8.

For a gamma-ray with incident energy $E$, the relationship between scattered photon energy $E'$ and angle $\theta$ is:

$$E' = \frac{E}{1 + \frac{E}{m_0c^2}(1 - \cos \theta)}$$ (2.3)
where \( m_0c^2 \) is the rest mass of the electron, 511 keV. Computing the probability of a given scattered energy requires using the Klein-Nishina equation (see [10], Section 2.3), but this is not necessary for this discussion.

In a spectrum, the energy recorded by the detector is approximately the energy that is deposited in the detector, which is the recoil electron energy \( E_e' \).

In Equation 2.3, note that the recoiled electron energy is bounded by:

\[
E_e'(\theta = 0) = 0 \\
E_e'(\theta = \pi) = E - \frac{E}{1 + 2 \frac{E}{m_0c^2}}
\]

An idealized Compton continuum is shown in Figure 2.9. At \( \theta = \pi \), the recoil electron energy is at a maximum, and a sharp dropoff called the Compton edge is observed. However, this treatment assumed that the secondary scattered photon escapes the detector (otherwise, the full photon energy can still be deposited in the detector) and that only one scattering event occurs. In the case of multiple scattering events (more significant for larger detectors and higher energy photons), higher order continua sometimes can be seen, although this is generally difficult to observe with low-resolution detectors and typical field scenarios. The sharp cutoff in the Compton continuum at \( \theta = \pi \) is called the Compton edge.

A measured spectrum of \(^{137}\)Cs is shown in Figure 2.10. The first-order Compton edge can be seen at 478 keV, and while the second-order Compton edge occurs at 555 keV, it is not readily detectable in this measurement. However, there are still other features in this spectrum that are yet to be described.

2.3.3 Bremsstrahlung Continuum

Any source that emits beta particles (or creates them via pair production) will have a bremsstrahlung continuum as well, though this is only significant with beta energies greater than 1 MeV [11]. Because betas are charged particles (either electrons or positrons), they decelerate as they travel through matter via electromagnetic interactions. The kinetic energy lost by the betas
Figure 2.9: An idealized Compton continuum [11], where only one scatter event is allowed.

is released as x-rays, called bremsstrahlung (which is German for “braking radiation”). This process creates a continuous spectrum of x-rays, as demonstrated in Figure 2.11.

2.3.4 Peaks From Other Processes

Aside from the direct gamma-ray emission described previously, there are other processes that create a peak in a spectrum. These processes include pair production, Compton scattering, and other nuclear reactions.

2.3.5 Pair Production

When a gamma-ray with energy greater than $2m_e c^2 = 1022$ keV interacts with the strong electric field around a nucleus, the spontaneous creation of an electron-positron pair can occur. By energy conservation, any leftover energy from the pair production is shared by the electron and positron in the form of kinetic energy. For the gamma-ray energies relevant to this dissertation (under 3000 keV), the electron and positron travel at most a few millimeters before losing their kinetic energy via interactions with the strong electromagnetic fields around nuclei. For example, $^{18}$F emits a positron with
a maximum energy of 634 keV, which has a range of 0.17 cm in plastic and 0.1 cm in aluminum [37]. When the positron annihilates with an electron, two 511 keV gamma-rays are produced, which have a significantly larger range.

For the observed spectra, pair production manifests several features. First, if one of the annihilation photons escapes and all of the other photons are absorbed by the detector, the total energy deposited in the detector is the original gamma-ray energy minus 511 keV; the resulting peak in the spectrum is the \textit{single escape peak}. If both of the annihilation photons escape, a \textit{double escape peak} with energy 1022 keV less than the original gamma-ray is observed. Finally, the annihilation process can occur outside of the detector; the absorption of a single annihilation photon produces a wide peak at 511 keV (the width is due to the Doppler shift). These can be seen in the spectra of $^{232}$Th in Figure 2.12.

For the problem of isotope identification, the effects of pair production can be negated by comparing the locations of the observed peaks. Because single or double escape peaks are always 511 or 1022 keV lower respectively than the photopeak, are wider than true peaks, and only occur for higher energy peaks, they can generally be discerned from true peaks easily.
2.3.6 Backscatter Peak

In addition to the continuum effect, Compton scattering outside of the detector volume also leads to the formation of a backscatter peak [10]. In Equation 2.3, consider the limit of $E \gg m_e c^2$ and suppose that $\theta \approx \pi$ (hence “backscatter”, i.e. the photon scatters backwards from the electron). Then:

$$E' = \lim_{E \gg m_e c^2/2} E = \lim_{E \gg m_e c^2/2} \frac{E}{1 + \frac{E}{m_0 c^2} \left(1 - \cos \theta\right)}$$

$$= \frac{E}{m_0 c^2 (1 - \cos \theta)}$$

$$= \frac{m_0 c^2}{1 - \cos \theta}$$

$$\approx \frac{m_0 c^2}{2} = 256 \text{ keV}$$

In Figure 2.10, the backscatter peak can be seen at 184 keV; this differs in energy slightly from the above because the 662 keV gamma-ray emitted in the decay of $^{137}\text{Cs}$ is not suitably large for the limit to hold. For the problem of isotope identification, this peak sometimes causes issues. While it is not Gaussian-shaped, many peak-detection algorithms will still identify it as a peak. Further, many isotopes of interest such as $^{235}\text{U}$ have important

Figure 2.11: The bremsstrahlung spectrum from the beta decay of $^{210}\text{Bi}$, with a maximum energy of 1161.3 keV [36].
emissions in this energy region (e.g. 185.7 keV for $^{235}$U), and the presence of this backscatter peak may obfuscate their signal.

2.3.7 Other Nuclear Processes

Other forms of radiation can cause secondary emission of gamma-rays. In particular, thermal neutron absorption by hydrogen causes a 2.2 MeV gamma-ray via $^1$H$(n, \gamma)^2$H. This interaction is especially useful because very few isotopes of interest emit gamma-rays near this energy, so a 2.2 MeV peak is a strong indicator of the presence of a neutron source. An example of this is shown in Figure 2.13; here, a sphere of weapons-grade plutonium (the “BeRP” ball, a 4.5 kg sphere of beryllium-reflected plutonium, 93.73% $^{239}$Pu, 5.97% $^{240}$Pu, and 557 ppm $^{241}$Am [38]) is surrounded by a layer of polyethylene (a hydrogen-rich plastic), and a wide peak can be seen at 2.2 MeV.
Figure 2.13: A 300 second spectrum of the BeRP ball (a WGPu source) encased in 4 cm of polyethylene, with the detector placed 1.2 m from the center of the source. A wide peak at 2.2 MeV can be seen, although its area is low despite the measurement time.

2.3.8 Shielding

One of the hardest problems for nuclear isotope identification is shielding. Gamma-rays (and other radiation) are attenuated by intermediate materials. This attenuation is a combination of scattering and absorption processes; the overall effects will be discussed here.

A monoenergetic beam of radiation with intensity \( I(x) \) and energy \( E \) is attenuated approximately as [39]:

\[
I(x) = I(0)e^{-\mu x}
\]

where \( \mu \) is the total linear attenuation coefficient and \( x \) is the distance traveled in a medium. Note that ignoring detector efficiency, if \( I(x) \) is measured at the surface of the detector, then the total counts in the spectrum would be equal to \( I(x) \) integrated over time. However, the total linear attenuation coefficient is a function of the shielding material and the energy of the gamma-ray; \( \mu \) is shown for a few materials in Figure 2.14.
As shown above, the linear attenuation coefficient $\mu$ is much larger at lower energies; that is, a gamma-ray with lower energy is much more likely to interact than a higher energy gamma. In observed spectra, low-energy or small peaks may not be resolvable if any shielding is present. In the $^{152}\text{Eu}$ spectra with different shielding configurations in Figure 2.15, the peaks at lower energies are no longer visible. Also, since gamma-rays at all energies are attenuated, the total counts observed in the spectra differ between the shielded and unshielded cases; with more shielding, one could completely hide the presence of a gamma-emitting source.

For the detection scenarios discussed in this dissertation, the shielding materials are one set of the many unknown variables. How this affects other identification algorithms is discussed in Chapter 3. Overcoming this challenge for the Bayesian classifier method of this dissertation is discussed at length in Chapter 5, particularly in Section 5.4.5.
In general, a single peak area does not give much information about the shielding materials and the radiation source, particularly for peak-based analyses. Suppose a fixed-position source with activity $\lambda(t)$ that emits gamma-rays with energy $E$ and branching ratio $BR(E)$ is placed $r_s$ away from a detector. Suppose the detector face has radius $r_d$ and intrinsic efficiency $\epsilon(E)$, and the measurement is performed for $T$ seconds. Finally, suppose there is some attenuating material with attenuation coefficient $\mu(E)$ and thickness $d$. Then the expected peak area $A$ is:

$$ A = \int_{t=0}^{T} \frac{4\pi r_d^2}{4\pi r_s^2} \cdot \lambda(t) \cdot BR(E) \cdot e^{-\mu(E)d} dt $$

(2.8)

However, the observed peak area is actually a Poisson random variable. Further, this assumes that only one attenuating material is present, that the source position and the detector positions are fixed, that the source has no self-attenuation, and that all of the quantities are known. For typical field-usage scenarios, many of these values (distances, activities, source) are not known, so this cannot be used directly. The method presented in this dissertation removes the need to observe for these values, as discussed in Chapter 5.

Figure 2.15: Spectra of $^{152}$Eu, without shielding and with 2.1 cm of Pb.
2.4 Other Radiation Sources

Finally, the radiation source of interest is rarely the only radiation source contributing to a spectrum. In most cases, a terrestrial radiation background and a cosmic background is present.

Naturally-occurring radioactive materials (NORM) include $^{40}\text{K}$, $^{232}\text{Th}$, natural uranium (99.3% $^{238}\text{U}$ and 0.7% $^{235}\text{U}$), $^{222}\text{Rn}$, and their decay chains. These materials are significant components of dirt and rock, and they vary in concentration geographically. A concentration map of a few NORM isotopes is shown in Figure 2.16. The contribution of NORM isotopes to a spectrum is unfortunately not temporally static. Weather conditions will strongly affect the NORM signatures; for example, rain will cause an increased contribution from radon, as it leaches from the soil in wet conditions.

Outside of the Earth’s atmosphere, many processes create extremely high energy radiation, some of which eventually reaches Earth’s atmosphere. While this radiation does include gamma-rays, it primarily consists of highly en-
ergetic charged particles. When this radiation interacts in the atmosphere, it creates a shower of secondary radiation. The contribution of cosmic radiation to a spectrum measured by a handheld detector (presumably on or near the Earth’s surface) is a function of latitude, longitude, elevation, and space weather conditions. As shown in Figure 2.17, elevation and latitude are very strong predictors of cosmic radiation levels. This primarily manifests in measurements as 511 keV gamma-rays (and the associated continuum effects) from the annihilation of positrons from pair production, but at high elevations a significant bremsstrahlung continuum can be observed [41].

Beyond the terrestrial and cosmic background, many industrial processes and some plants produce materials with elevated radioactivity. Technologically enhanced naturally-occuring radioactive materials (TENORM) are present in virtually all developed areas. Building materials such as granite or brick contain elevated \(^{238}\text{U}\), and \(^{232}\text{Th}\) levels; food and tobacco products have elevated \(^{40}\text{K}\) and \(^{226}\text{Ra}\) concentrations due to the fertilizer components, and so on [16].

There are many approaches for accounting for radiation background, some of which are limited to specific isotope identification algorithms and are discussed in Chapter 3. The simplest of these is to perform simple background subtraction. A spectrum is measured with the source, and then repeated under the same conditions but with the source removed. Then, as shown in Figure 2.18, the source signal is then the total signal minus the background signal. However, for many of the usage scenarios for these handheld identifiers, it is not practical or even possible to remove the source, as the source could be an unknown object, inaccessible, or distributed (as in spilled waste scenarios). Second, the number of counts in a channel is a Poisson random variable, so the background subtraction may lead to negative counts in some channels. For the algorithm developed in this dissertation, no background reduction is performed, although it certainly can be done without harming performance (see Chapter 6).

There are also other radiation sources that would generally only be temporarily present, often making it impossible to subtract off their contribution to the spectrum. When patients undergo radiotherapies, they become mobile radiation sources for days to weeks afterwards. For example, radiation therapy for thyroid cancers often includes the administration of \(^{131}\text{I}\) (with an effective half-life of 4.6 days [42]), and treatments for neuroendocrine tumors
Figure 2.17: Top: An elevation map of the United States. Bottom: a map of cosmic-ray exposure [40]. Note that the cosmic exposure rate is very strongly correlated with altitude and latitude.
Figure 2.18: Example of background subtraction on gamma-ray spectra. Top: 30 minute spectrum of a 0.3 $\mu$Ci $^{137}$Cs source, measured from 30 cm. Middle: 10 minute background measurement of Talbot Laboratory 209D, scaled for direct subtraction. Bottom: the background-subtracted $^{137}$Cs spectrum. Note that the background subtraction does not perfectly eliminate counts in the higher energy channels due to Poissonian noise in all measurements. In this final spectrum, the 1460 keV peak of $^{40}$K and the 2614 keV peak from the $^{232}$Th decay chain are no longer visible.

may include $^{177}$Lu. Other common medical isotopes such as $^{99m}$Tc are used for medical imaging. These isotopes generally have half-lives on the order of hours to days to reduce the radiation dose delivered to the patients [42]. An isotope identifier needs to be able to distinguish between medical sources and other sources of interest; it is not practical or ethical to have to detain every person that has undergone medical treatment for additional radiation screenings.

2.5 Summary

Shielding materials, unknown radiation sources, detector resolution and efficiency limitations, and more all obfuscate spectra. In the field of isotope
identification, we seek to identify radioisotopes in a wide variety of circumstances; information about intermediate materials, geometry, background radiation, etc. is generally unavailable. A spectroscopist or an isotope identification algorithm must consider all of the features of a gamma-ray spectrum if reasonable performance is to be achieved.
3.1 Introduction

Once a spectrum has been measured, there are many methods to actually perform isotope identification. This chapter provides a general overview of the categories of algorithms that are used for this problem, as well as the motivation for algorithm improvement.

One of the most recently published evaluations of commercial handheld isotope identifiers is [43] from 2004. This evaluation used seven different commercial identifiers on 443 measurements, and the results are shown in Figure 3.1. For each measurement, the most abundant isotopes (MAI) were known, and for each measurement the identification was labeled as one of the following:

- **Correct**: at least one MAI was identified correctly
- **Conditionally correct** (the MAI was identified but with less confidence than an incorrect isotope)
- **Minor daughter**: the MAI was not identified, but another isotope in its decay chain or a contaminant isotope (< 10% abundance) was identified instead.
- **False negative**: No isotopes were identified, or only a background isotope (e.g. $^{40}$K) was identified.
- **False positive**: The instrument wrongly reported one or more radionuclides without making any correct identification.

They found that the identifiers only made correct or conditionally correct identifications in approximately a third of all measurements and even less
on special nuclear materials (SNM), industrial, and medical sources. The false negative rate was nearly the same as the correct or partially correct identification rate.

While identification algorithms have almost certainly improved in the past 12 years, updated detector evaluations have not been published, due to both the proprietary nature of the algorithms and the need to not reveal specific security gaps.

The algorithms implemented on commercially-available RIIDs are proprietary, so it is unknown what specific identification methods are being used on-board [6]. This complicates the issue of algorithm development and evaluation; a new program through the Department of Homeland Security (DHS) seeks to improve this situation and is discussed later in Chapter 6.

Most methods for passive isotope identification fit into one of a few categories: library comparison, region of interest (ROI), template matching, eigenvector methods (such as PCA), neural network approaches, or expert interaction methods [7]. However, expert interaction methods are not a fully automated method in implementation and are not practical for the majority of handheld RIID usage, so they will not be considered here.

3.2 Library Comparison Algorithms

Library comparison (a.k.a. peak matching) methods compare the measured peak energies in the data against a library of isotopes. This method is gen-
Figure 3.2: An example of a poor calibration due to gain shift. The measured spectrum is shown in black, while the blue overlay is the library entry. Reproduced from [16]. The discrepancy between measured peak energies and expected peak energies presents a problem for library comparison algorithms.

erally the simplest class of ID algorithm, but it is the most susceptible to issues with NaI detectors. Because this method simply compares the measured centroid energies in the data against an isotope library, poor detector calibrations and gain shifts (see Figure 3.2) ruin its performance [7]. It is also completely reliant on accurate peak localization/extraction from a spectrum, which can be difficult with handheld NaI detectors or with low-count spectra. Poor counting statistics will cause missed peaks or errors in the peak fitting, as will inaccurate detector calibrations. An example of the effects of counting statistics is shown in Figure 3.3. For more complicated sources with a large number of photopeaks, many peaks are not detectable unless the spectra are measured for long periods of time.

This class of algorithm can also be fooled by similar-appearing isotopes, such as $^{177m}$Lu and $^{239}$Pu. These isotopes have many similar gamma-ray emissions, but the branching ratios are different. In particular, $^{177m}$Lu has peaks at 378.5 keV and 413.7 keV, while $^{239}$Pu has peaks at 375.0 keV and 413.7 keV, among others; a larger list of gamma-rays is shown in Table 3.1. If identification is performed without using area information, it will be difficult
Figure 3.3: The effect of counting statistics on measured spectra of a 0.3 μCi $^{133}$Ba source. As more gamma-rays are detected, more photopeaks become detectable, e.g. the 1460 keV peak from background $^{40}$K.

or impossible to distinguish between the two. This example is particularly problematic, as one is a common medical isotope, and the other is highly-controlled SNM.

Likely due to the simplicity of this method, this algorithm class is underrepresented in published literature. One example implementation of this algorithm is on the GR-135 detector [34] (note that there have been updated versions of this identifier, so this method may not be used any more). Here, peaks are detected and fit with an unspecified method. Peaks with areas below a predefined threshold are removed from consideration; the threshold is set simply as a percentage of the total peak areas. Note that this method will likely create issues when dealing with high energy peaks, as they generally have areas that are orders of magnitude below low energy peaks.

Next, the peak data is compared against an onboard library to produce identifications. There are multiple ways that this could actually be performed. The simplest method would be to have a list of peaks for each isotope that should be detected if it is present; if all (or enough) of the peaks are identified, then that isotope would be identified. This breaks down, however, as many isotopes have peaks at similar energies, and this would produce multiple identifications for a single source.

A better method would be to find which isotope explains most of the data, then see which isotope explains the most of the remaining unexplained data,
Table 3.1: Comparison of selected major gamma-ray emissions of $^{177}$Lu and $^{239}$Pu. These isotopes have 66 and 157 total gamma-ray emissions respectively.

<table>
<thead>
<tr>
<th>Energy</th>
<th>BR</th>
<th>Energy</th>
<th>BR</th>
</tr>
</thead>
<tbody>
<tr>
<td>55.790</td>
<td>58.500</td>
<td>38.674</td>
<td>5.86E-03</td>
</tr>
<tr>
<td>63.166</td>
<td>19.100</td>
<td>51.622</td>
<td>2.08E-02</td>
</tr>
<tr>
<td>112.949</td>
<td>21.500</td>
<td>98.434</td>
<td>5.90E-03</td>
</tr>
<tr>
<td>128.497</td>
<td>15.200</td>
<td>111.025</td>
<td>2.25E-03</td>
</tr>
<tr>
<td>153.290</td>
<td>17.800</td>
<td>129.283</td>
<td>6.20E-03</td>
</tr>
<tr>
<td>174.403</td>
<td>12.700</td>
<td>171.370</td>
<td>1.09E-04</td>
</tr>
<tr>
<td>208.363</td>
<td>60.900</td>
<td>203.520</td>
<td>5.60E-04</td>
</tr>
<tr>
<td>281.787</td>
<td>14.100</td>
<td>255.340</td>
<td>8.03E-05</td>
</tr>
<tr>
<td>319.028</td>
<td>11.000</td>
<td>332.810</td>
<td>5.05E-04</td>
</tr>
<tr>
<td>378.502</td>
<td>27.700</td>
<td>375.018</td>
<td>1.58E-03</td>
</tr>
<tr>
<td>413.658</td>
<td>17.500</td>
<td>413.691</td>
<td>1.51E-03</td>
</tr>
<tr>
<td>418.530</td>
<td>20.100</td>
<td>422.570</td>
<td>1.19E-04</td>
</tr>
</tbody>
</table>

and so on. This is still going to fail often for several reasons. First, it could be the case that several simple isotopes are present, but this method would try to force an identification of more complex isotopes. Second, when dealing with low-resolution spectra, the presence of multiple isotopes will distort peaks non-linearly; this effect is discussed in detail in Section 4.3.

Finally, the detector could consider all of the peaks that were not accounted for in the nuclide identification process. The total area of the unexplained peaks is compared against the total area of all peaks; if this ratio is above a threshold, an identification of “Unknown Isotope” is added to the final decision. This step is included in commercial detectors such as the GR-135 (version 1V22).

3.3 ROI Algorithms

ROI methods look at the total counts in small regions of a spectrum and compare it against the background count rates in those regions. Identifications are made by comparing the regions with elevated counts against the expected elevated count regions for different isotopes. Identifications can be made a few ways from these ROIs. One method would be to alarm for a certain isotope if all of its predefined ROIs have elevated counts. Another
method would be to examine the ratio of elevated ROIs and compare them against a given isotope.

This method is extremely computationally inexpensive, but its nature makes having extensive isotope libraries infeasible, as many isotopes have overlapping regions, making the determination of non-intersecting regions of interest impossible for more than a few isotopes of interest [7]. A common counter-example for this algorithm is the aforementioned $^{239}$Pu (significant peaks at 375.0 keV and 413.7 keV) and $^{177m}$Lu (significant peaks at 378.5 keV and 413.7 keV) [35]; while these peaks have different branching ratios and thus can be distinguished between by examining their areas, a simple ROI method does not contain this logic.

An example implementation of an ROI method is shown in Figure 3.4. While $^{133}$Ba and $^{239}$Pu both have peaks in this region, it is possible to set distinct regions for each for successful discrimination. However, this will become more challenging if more isotopes are to be considered. This also relies on accurate background subtraction and well-calibrated detectors. A gain shift of a few percent could shift the barium peaks into the same region as plutonium, destroying the identification. In the previous example of Figure 3.2, depending on how narrowly the ROIs are defined the algorithm may not trigger on $^{67}$Ga. The shift may also cause ROIs for other isotopes to alarm.

### 3.4 Decomposition and Deconvolution Methods

Decomposition and deconvolution methods seek to reduce the dimensionality of spectra [9, 45]. These techniques can be used either for pre-processing spectra, for identification purposes, and everywhere in-between.

Deconvolution techniques aim to improve the spectral data to create a “better” representation of the data for interpretation. The observed spectrum $O(E)$ is (ignoring noise) the convolution of the incident spectrum $I(E)$ and the detector response function $R(E, E_0)$ [45].

\[
O(E) = \int_0^\infty R(E, E_0) \cdot I(E_0) dE_0
\]

There are multiple approaches for finding the deconvolution here, such as linear regularization, Maximum Likelihood Estimation using Expectation Max-
Figure 3.4: Demo of an ROI method for discriminating between $^{133}$Ba (top) and $^{239}$Pu (bottom). Elevated counts in ROI 1 would signal $^{133}$Ba, while elevated counts in ROI 2 would alarm for $^{239}$Pu.

Of decomposition techniques, by far the most common used in isotope identification problems is principal component analysis (PCA) [24, 47, 48], which transforms the data into a new set of uncorrelated variables with a reduced dimensionality. By finding a new set of basis vectors to represent the data, the goal is that different identification classes will be well separated. This idea is not unique to PCA; other methods such as Fisher discriminant analysis attempt to do this as well [49].

In PCA, a large set of spectra (call the matrix of all the spectra $X$) is multiplied by a rotation matrix $P^T$ to try to get the data vectors to lie along the directions of maximum variation [50]. If the new data representation is $Y = P^T X$, the rotation matrix $P^T$ is chosen such that the covariance matrix of $Y$ is diagonal. With some straightforward linear algebra, it can be shown that the matrix $P$ should be the matrix of eigenvectors of $cov(X)$. In
Figure 3.5: An example of ML-EM deconvolution (bottom) using a simulated spectrum (top). Figures reproduced from [45].

In practice, $P$ is constructed with the eigenvectors ordered by the magnitude of the corresponding eigenvalues. Generally, eigenvalues below some threshold are rejected, which will reduce the dimensionality of the dataset.

To perform identification or classification, some discriminating function must be applied to separate the different classes. Whenever a new spectrum is measured, its distance from each of the clusters would be computed by some metric, e.g. Mahalanobis distance [24]. Ideally, the classes would be well-separated, but this is not generally the case for low-resolution spectra. A demonstration of PCA is shown in Figure 3.6; isotope identification is attempted with moving radiation detectors, but not all of the classes are well-separated [48].
3.5 Template Matching Methods

Template matching methods are a powerful and well-studied identification method for low-resolution detectors [51]. A vast library of spectra is recorded (or simulated) for many isotopes and configurations, and for a test spectrum an error metric, e.g. Mahalanobis distance or a $\chi^2$ test, is calculated to each library template. This has the effect of fitting the whole spectrum at once. With modifications, this can minimize the effect of poor calibrations and unknown gain drift. One of the most well-developed isotope identification codes is the Gamma Detector Response and Analysis Software (GADRAS), which uses a template-based approach for making identifications [52]. However, this software is currently official-use only (OUO) and is unavailable outside of governmental use, making its use in commercial RIIDs impossible.

There are other advantages to template matching methods other than raw isotope identification. Because the whole spectrum is being used, the shape of the continua are important for the optimal templates. The shape of these continua are a complex function of the source configuration and shielding geometry. With a large enough template library, more information about the configuration can be extracted. This is particularly important for applications like warhead verification [53].
3.6 Neural Network Approaches

There are many machine learning approaches that either partially fall into one of the previous categories or are entirely different. Generally speaking, machine learning methods haven’t been used much for RIIDs due to the enormous amount of data necessary to adequately train the algorithm and the difficulty in “teaching” an algorithm to understand general physical models. While there are many different algorithms within this family of methods, discussion for this dissertation is restricted to neural networks due to their presence in published literature and their development by the isotope identification research group at UIUC.

Artificial neural networks (ANNs or simply NNs) are a class of mathematical models that were originally developed to mimic biological neurons [54]. Only a simple feed-forward neural network is discussed at length here, but there are many different structures of and variations on neural networks in literature that have been applied to isotope identification. Published work includes not only these simple feed-forward networks [55, 56, 57, 58], but also variations such as abductive neural networks [59], radial basis function (RBF) neural nets [60], and convolutional neural networks [61]. Further, these methods have been applied to various related problems, such as peak fitting [62], isotope identification [59, 56], activity or enrichment estimation [59, 57, 60], and other similar problems [55, 63, 64, 65]. Of these, only very few are specific to isotope identification with low resolution detectors; these will be discussed at the end of this section.

As shown in Figure 3.7, a feed-forward neural network is composed of an input layer $A$, an output layer $C$, and some number of hidden layers (in this image, one hidden layer $B$ is shown). Each of these layers consists of a number of neurons (e.g. $B_1$, $B_2$, etc.), which are “activated” by passing a weighted sum of the previous layer’s output through an activation function, e.g. the sigmoid function $f(t) = 1/(1 + e^{-t})$. This activation is shown in Figure 3.8.

There are many methods to train a feed-forward network, such as gradient descent, simulated annealing, and others [49]. In general, a set of labeled training data is used to compute the accuracy of the network at labeling this dataset, and the parameters of the network are adjusted until the errors on this training set are minimized [49].
Of the publications mentioned earlier, few of the neural network applications have been specific to isotope identification. Of those, there are a common set of approaches and challenges. First, most do not use the full spectrum as an input to the neural network; instead, channels are either combined to reduce the input dimension significantly, or the spectra are processed in some way (e.g. PCA, or peaks are extracted first [67], etc.).

Second, most of these publications used HPGe detectors, which have a significantly better energy resolution than NaI (approximately 25x finer, see Figure 2.6) but at significantly higher cost and complexity in use. Only [68, 69, 70] used NaI detectors. These papers noted significant performance degradation when a gain shift (changing the energy calibration) occurred, the
source-detector geometry was changed, or an unknown source was added to the mix. These problems may be solvable for neural networks by significantly increasing the volume and variety of training data, but this has not yet been fully explored.

Third, these papers use a very limited library; the largest of these is [70], which used a library of 10 isotopes (Na-22, Mn-54, Co-57, Co-70, Cs-137, Eu-152, Eu-153, Eu-154, Ra-226, Th-232). To be appropriate for field use, a much larger library will be required. As discussed later in Chapter 4, the ANSI standard for handheld detectors requires 18 isotopes, and a more practical library will be even larger. This will present a significant challenge to neural network approaches due to the increase in training complexity and the dimensionality increase in the basis space.

These groups have considered neural network approaches for isotope identification, source classification, isotopic analysis, and other problems, but generally conclude that these approaches aren’t feasible due to computational limitations, training data limitations, or other difficulties. However, as computation becomes cheaper and better software libraries become available, neural network approaches become more feasible.

These challenges facing neural network approaches are likely not insurmountable. The majority of papers investigating their use are 10-20 years old, and computational resources have improved massively since then. For this reason, there are two projects within the RDII group at UIUC developing more sophisticated neural networks for feature extraction and isotope identification [66], rather than simple “threat” vs “non-threat” classification. Large training data sets can be simulated by the approach described in Section 4.4, eliminating one of the major challenges associated with this approach. High performance cloud computing is readily available for network training, potentially solving the other major challenge to this approach.

3.7 Other Comments on Existing Algorithms

While some of these methods are not suitable for low-resolution handheld detectors, they may be well-suited for other usage scenarios. For example, when working with a high-resolution detector like high-purity germanium, accurate peak detection is feasible [71] and even simple library comparison
methods can achieve greater accuracy. However, most handheld detectors use low-resolution technologies like sodium iodide due to cost and their stability in field scenarios. The poor resolution, combined with the complicating phenomena described in Chapter 2 and the limited computational capabilities onboard these handheld detectors, makes accurate isotope identification a significant challenge.

3.8 Proposed Identification Algorithm

To address the challenges presented by the problem of isotope identification on low-resolution detectors, we propose a new method based on feature extraction and Bayesian classifiers. Because a trained spectroscopist is capable of accurate identifications, this methodology was designed to mimic the spectroscopist’s identification process. This approach will use peak information to make identifications. This methodology can handle miscalibrated detectors and unknown shielding, and could be easily extended to account for other information sources (e.g. neutron measurements). A non-trivial isotope library based on the ANSI standard for handheld RIIDs [72] is used for early testing, though a larger library is currently being developed, as discussed in Section 4.3.

As discussed throughout the next section, this method is closest to a library comparison method but avoids most of the issues that ruin the performance of that class of algorithm. It is capable of using much more spectral and situational information, as well as being modular enough to enable constant upgrades and optimization. Demonstrations of the algorithm and performance benchmarks are presented in Chapter 6, and future development is outlined in Chapter 7.1.
4.1 Introduction

All isotope identification algorithms involve some sort of spectral “library”, where a library is the set of representations for each isotope [6, 7, 8]. For a template matching approach, this would be a large set of isotope templates; for peak-based approaches (as in this work, described in Chapter 5), this library is a set of peak energies and areas (or branching ratios) for each isotope (and each combination of isotopes, as described later in Section 4.3).

While a truly general purpose identification method would require an isotope library of approximately 200 radioisotopes [7], the mission space for handheld low-resolution identifiers has a significantly smaller requirement. These identifiers are primarily being used to detect and identify illicit material, particularly in places like shipping ports and national borders. The ANSI 2006 N42.34 standard for handheld isotope identifiers suggests the following isotopes [72]:

- **SNM**: $^{233}\text{U}$ and $^{235}\text{U}$, $^{237}\text{Np}$, Pu
- **Industrial**: $^{57}\text{Co}$, $^{60}\text{Co}$, $^{133}\text{Ba}$, $^{137}\text{Cs}$, $^{192}\text{Ir}$, $^{204}\text{Tl}$, $^{226}\text{Ra}$, $^{241}\text{Am}$
- **Medical**: $^{18}\text{F}$, $^{67}\text{Ga}$, $^{51}\text{Cr}$, $^{75}\text{Se}$, $^{89}\text{Sr}$, $^{99}\text{Mo}$, $^{99m}\text{Tc}$, $^{103}\text{Pd}$, $^{111}\text{In}$, Iodine ($^{123}\text{I}$, $^{125}\text{I}$, $^{131}\text{I}$), $^{153}\text{Sm}$, $^{201}\text{Tl}$, $^{133}\text{Xe}$

However, the actual requirements are that the following 18 isotopes be identifiable within a maximum of two minutes, where the detector is placed such that the exposure rate at the detector is 50 $\mu\text{R}/\text{h}$ [72]:

- **Unshielded**: $^{40}\text{K}$, $^{57}\text{Co}$, $^{60}\text{Co}$, $^{67}\text{Ga}$, $^{99m}\text{Tc}$, $^{125}\text{I}$, $^{131}\text{I}$, $^{133}\text{Ba}$, $^{137}\text{Cs}$, $^{192}\text{Ir}$, $^{201}\text{Tl}$, $^{226}\text{Ra}$, $^{232}\text{Th}$, $^{233}\text{U}$, $^{235}\text{U}$, $^{238}\text{U}$, Pu (reactor grade, > 6% $^{240}\text{Pu}$), $^{241}\text{Am}$

40
- Shielded (behind 5 mm steel shielding): $^{40}$K, $^{57}$Co, $^{60}$Co, $^{67}$Ga, $^{99m}$Tc, $^{125}$I, $^{131}$I, $^{133}$Ba, $^{137}$Cs, $^{192}$Ir, $^{201}$Tl, $^{226}$Ra, $^{232}$Th, $^{233}$U, $^{235}$U, $^{238}$U, Pu (reactor grade, > 6% $^{240}$Pu), $^{241}$Am

Finally, the ANSI requirement also stipulates that the detector must be able to simultaneously identify a mixed source containing $^{133}$Ba and reactor grade Pu, with the same two minute and 50 µR/h requirement.

For each of these sources, the test must be repeated ten times, and the detector must make correct identifications for eight out of the ten trials for each source. There are other requirements for these detectors, but they are outside the scope of this dissertation (e.g. hardware requirements, etc) [72].

In Chapter 6, algorithm performance will be demonstrated using a library containing the ANSI required isotopes. This library contains a reduced set of peak energies and areas for each of these isotopes. Before describing the method for library reduction, the feature extraction method must first be described, as it is coupled to the library generation routine described in Section 4.3.

### 4.2 Feature Extraction

To perform automated peak-based identification, a peak detection and quantification algorithm must be implemented. For this work, a wavelet/non-negative least squares (NNLS) method was used to detect and fit peaks in a spectrum [73, 74, 75, 76]. While this algorithm is important to the implementation of the identification algorithm presented in this dissertation, the wavelet/NNLS code is only described briefly here, as it is generally outside the scope of this work.

A wavelet $\psi(t)$ is a zero-area, square-integrable signal that is non-zero over a finite domain. An example wavelet (the ‘bior2.6’ wavelet by MATLAB’s naming convention, a member of the biorthogonal family [77]) is shown in Figure 4.1. The wavelet is shifted by parameter $E$ and scaled by parameter $a$, as

$$\psi_{E,a}(t) = \frac{1}{\sqrt{a}} \psi \left( \frac{t - E}{a} \right) \quad (4.1)$$
The continuous wavelet transform (CWT) of a signal $f(t)$ is the convolution integral of the signal and the wavelet.

$$T(E, a) = \int_{-\infty}^{\infty} f(t) \cdot \frac{1}{\sqrt{a}} \psi^\ast(\frac{t - E}{a}) dt$$ (4.2)

By computing the CWT of a spectrum over a range of scale and shift parameters, peak information can be extracted. Consider the perfectly Gaussian signal and its CWT in Figure 4.2. Maxima in the CWT are colocated with the peak. By finding the local maxima over the shift parameter, wavelet transform modulus maxima (WTMM) lines are formed. Finally, finding maxima along the WTMM lines gives peak width information.

In reality, some of the WTMM lines in a CWT are not useful for peak detection. There are many approaches for filtering out these unneeded maxima [75]. Most importantly, a photopeak in a spectrum that was collected with an NaI detector has a width that is determined by the photopeak’s energy. Because this relationship is known, an optimal scale curve can be computed.
Figure 4.2: Top: a Gaussian signal. Bottom: The wavelet transform of the signal. The WTMM line (solid) and the maximum along the WTMM line (white ‘x’) give information about the peak centroid and peak width. Comparing the scale at maximum transform value against the optimal scale curve (dashed) can provide information on potentially overlapping peaks and the cause of a peak [73].

The optimal scale $\alpha(E)$ is the scale that produces a maximum in the CWT of a perfectly Gaussian photopeak with mean $E$ and full-width at half-max (FWHM) that matches the detector’s FWHM versus $E$ curve. This optimal scale curve is overlaid on Figure 4.2.

To get better estimates of peak centroid and to obtain area information, non-negative least squares fitting was performed with a predetermined basis matrix. However, the NNLS was more easily defined using the tensor formulation of the wavelet transform approach. Let $S$ be the (1024x256) matrix of values from the CWT of $X$. The first dimension is chosen to match the length of a spectrum vector; the NaI detectors used here typically have a 1024-channel multichannel analyzer. 256 scales are chosen arbitrarily; in future work, this will be reduced to simplify the computations needed. Then the wavelet transform can be expressed as

$$WX = S$$

(4.3)
where $X$ is the spectrum (1024x1 matrix), and $W$ is the wavelet transform tensor (1024x256x1024).

Next, define:

- $G_j$ = a normalized (area equals one) Gaussian function with centroid (mean) in channel $j$ and a standard deviation that matches the full width at half-max (FWHM) response of the detector. A 1024x1 vector.

- $[CWT(f)] = \text{the CWT of signal } f \text{ along the optimal scale only. A 1024x1 vector.}$

- $B_{ij} = [CWT(G_j)]_i$, the basis matrix. For this work, $B$ is a 1024x256 matrix [78].

Given the basis matrix $B$, NNLS is performed to find the fit vector $k$, whose nonzero elements ideally represent the identified peaks. In NNLS, a submatrix of $B$ is defined by setting some columns of the basis matrix to zero to minimize the fit vector error subject to the nonnegative constraint. Let $B_1$ be this submatrix of $B$.

\begin{align*}
Bk &= S \tag{4.4} \\
B_1^T B_1 k &= B_1^T S \tag{4.5}
\end{align*}

If the inverse exists, the solution can be written explicitly as

\begin{align*}
k &= (B_1^T B_1)^{-1} B_1^T S = OS \tag{4.6} \\
O &= (B_1^T B_1)^{-1} B_1^T \tag{4.7}
\end{align*}

However, the matrix $B_1^T B_1$ is singular. To avoid this issue, truncated singular value decomposition (TSVD) is to find a pseudoinverse. For notation, let $A^*$ denote the conjugate transpose of matrix $A$. For a matrix $A$, the pseudoinverse is the unique matrix $A^+$ that satisfies the following four equations [79]:

\[44\]
In TSVD, $B_1$ is decomposed as $UEV^T$, where $U$ and $V$ are unitary matrices and $E$ is a diagonal matrix of non-negative real numbers. The pseudoinverse of $B_1$ is then $VE^+U^T$, where $E^+$ is the pseudoinverse of $E$. The diagonal elements of $E^+$ are simply the reciprocal of the corresponding elements in $E$. In TSVD (versus the usual singular value decomposition), these diagonal elements are set to zero if they are below a user-defined threshold.

$$E^+_{i,i} = \begin{cases} E^{-1}_{i,i}, & \text{if } E^{-1}_{i,i} \geq 0.1 \\ 0, & \text{otherwise} \end{cases}$$

Finally, the fit vector is computed by:

$$B_1 = UEV^T \quad (4.12)$$
$$k = VE^+U^TS \quad (4.13)$$

To augment the isotope identification algorithm, the variance of the fit vector $k$ is computed. This information can be used as a peak filter (e.g. reject peaks with errors over a threshold) or as additional information in an isotope identification algorithm (e.g. weight peaks with a lower uncertainty more than high uncertainty peaks in a peak scoring algorithm). Let $C_S$ be the covariance of the coefficient matrix $S$ and $C_X$ be the covariance of the signal $X$. If $S$ is uncorrelated, $C_S$ can be calculated simply as:

$$C_S = \sigma_s^2 I \quad (4.14)$$
$$\sigma_s = \frac{1}{m-n}S_{obj} \quad (4.15)$$
$$S_{obj} = S^T(I-B_1O)S \quad (4.16)$$
Figure 4.3: A 600 s 0.25 µCi $^{60}$Co spectrum, measured from 10 cm, and the scalogram of the wavelet transform. Notice that the WTMM maxima occur above the optimal scale curve for the two photopeaks at 1173 and 1332 keV.

where $m - n$ is the degrees of freedom of the submatrix $B_1$.

However, $C_S$ is generally not uncorrelated and will be a function of the covariance of the signal $C_X$. Suppose $C_X = I\sigma_n^2$. Then:

$$C_S = WC_sW^T$$  \hspace{1cm} (4.17)

$$C_k = OC_sO^T$$  \hspace{1cm} (4.18)

and the variance of vector $k$ is $diag(C_k)$.

The results of the wavelet transform/NNLS code on a measured $^{60}$Co spectrum are presented in Figure 4.3. Due to the overlap of the 1173 and 1332 keV photopeaks, the maxima in the WTMM chains occur higher (at a smaller scale) than the optimal scale curve.
4.3 Isotope Library

To construct an appropriate isotope library, we begin with the complete set of peak energies and branching ratios for each model (isotope), which can be easily found in a database such as [35]. Throughout this dissertation, the term “model” will be used to mean an entry in this library. A model is the set of peak energies and areas for a given isotope or combination of isotopes.

To allow for the consideration of sources with multiple isotopes, combinations of isotopes are also stored directly in the library. However, using this raw library with low-resolution detectors causes issues with performance for peak-based identification methods.

To demonstrate this, consider the isotope $^{152}$Eu, whose spectrum and library is shown in Figure 4.4. This isotope has 156 photopeaks between 5 keV and 1800 keV. In the unshielded spectrum in Figure 4.4, perhaps a dozen peaks might be visible; adding a small amount of shielding reduces this even further, as shown in Figure 6.7. Clearly, a low-resolution detector will never be able to resolve most of these peaks, even with a long measurement. For identification algorithms, this complicates the issue of figuring out which measured peak corresponds to which library peak, especially if the spectrum is not calibrated perfectly. In cases like $^{152}$Eu, even a 1% calibration shift would cause many peaks to be mismatched, massively degrading identification performance.

The problem with excessive peak data is made significantly worse by the mixed sources stored in the library as well. It is clear, then, that some method must be used to reduce the library to remove overlapping peaks and peaks that are too small to be reasonably detectable. One method to eliminate overlapping peaks is as follows [74]:

- For a given feature extraction method (in this case, the wavelet/NNLS method), find the minimum resolvable energy (MRE); the MRE is the closest in energy two neighboring peaks can be and still be resolvable by the feature extraction.

- For each model in the library, find any peaks that are closer in energy than the MRE. Replace these peaks in the library with an “effective” photopeak (defined in Figure 4.6).

- Repeat until no unresolvable peaks are left.
Figure 4.4: The spectrum of $^{152}$Eu, overlaid with a stem plot of the complete library. Each of the 156 stems in the stem plot represents a photopeak, whose height is proportional to the branching ratio. Clearly, most of the peaks of $^{152}$Eu are not visible in the spectrum.

However, this approach has two problems. First, it does not eliminate excessively small peaks from the library. For the $^{152}$Eu example, this approach reduces it to approximately 30 peaks, which is still far too many to be appropriate for these detectors. Trying to eliminate small peaks from the library separately is difficult, as finding a threshold function that works for all isotopes across the full energy range is a challenge. Second, the energies of the effective photopeaks do not always agree with experimental observation due to the pair-wise method used for eliminating overlapping peaks. For this approach, a cluster of small overlapping peaks can bias the effective peak calculation away from a larger peak, and the resulting library is not the same if the pairs are formed from the left as from the right.

To solve these issues, a simple new method has been proposed to generate appropriate libraries. For each raw model in the library:

- Generate a “pseudospectrum” for this model; this is an ideal simulated spectrum with perfect counting statistics, no background radiation, and no attenuation from intermediate materials.
Figure 4.5: An $^{152}$Eu spectrum, where a small amount of lead shielding has been placed between the detector and the source. Notice how several of the lower energy peaks have been attenuated out by the shielding material.

Figure 4.6: When two Gaussian peaks (solid lines) have significant overlap, their addition yields an “effective” photopeak (dashed line) that is approximately Gaussian. Photopeaks are approximately Gaussian, so overlapping photopeaks can be approximated as a Gaussian mixture. When these peaks are so close together that they can’t be resolved, a good approximation of the Gaussian mixture is a single Gaussian with a new centroid and area, which is the “effective” photopeak.

- Use the feature extraction algorithm to detect and quantify peak information.
- Use the extracted information as the reduced model.

Each of these steps are discussed in the following subsections.

4.3.1 Generate a Pseudospectrum

Two different methods have been implemented for generating pseudospectra. The first method was to ignore all continua effects and scattering, and assume that the detector response to a monoenergetic gamma-ray source is a perfect Gaussian peak. For this approach, suppose that model $M_i$ originally contains the peak energies and branching ratios $\{BR_j\}_{j=1}^n$. Let the detector efficiency function be $\epsilon(E)$ and the detector variance function be $\sigma(E)$. Then the peak at $E_j$ will have the effective branching ratio $BR_j \ast \epsilon(E_j)$. This excessively idealized pseudospectrum was then expressed as:

$$PseudoSpectrum(E) = \sum_{j=1}^{n} \frac{BR_j \epsilon(E_j)}{\sqrt{2\pi\sigma(E)}} \exp\left(-\frac{(E - E_j)^2}{2\pi\sigma(E)^2}\right) \quad (4.19)$$

This pseudospectrum doesn’t have any of the continua effects that a true spectrum would have. Because there is no Compton continuum, smaller peaks are somewhat larger than would typically be expected (the peak area to Compton area ratio is an increasing function of energy due to the energy dependence of the scattering cross-section, described in Chapter 2).

A different method for pseudospectrum generation is to use the full spectrum simulation approach described in Section 4.4. This method has the advantage of generating more realistic spectra, complete with Compton continua, as demonstrated in the following section. However, the current feature extraction code sometimes misidentified Compton edges as photopeaks, and also sometimes missed photopeaks when they have significant overlap with a Compton edge. This lead to some incorrect peak data in the library (either nonexistent peaks or missing peaks that should be included), and overall worse identification performance in the final identification algorithm.
4.3.2 Extract Peak Information

To obtain peak centroid, area, and area uncertainty information from the pseudospectrum, the wavelet/NNLS code is used. Because there are many models in the library, the feature extraction code must process many spectra (between 50,000 to 250,000 depending on the library used). To be able to generate and regenerate libraries as needed, this step was performed on the UIUC Taub Cluster. This reduces the needed computation time from approximately one week to five hours.

4.3.3 Library Generation Output

A comparison of this method against the previous method is shown in Figure 4.7.

This approach has several significant advantages over the old method.

First, this method easily and automatically couples the detector to the library. If a new detector was to be used (e.g. a different sized NaI detector or a CsI detector), a new library can be easily generated using the characteristics of the detector (the efficiency and resolution curve). To use the old method for a new detector or for a new feature extraction algorithm, one would have to manually compute the MRE function.

Second, this method couples the feature extraction algorithm to the library. If, for example, a replacement to the wavelet/NNLS code was to be used or if the wavelet/NNLS code was modified significantly, a new library can be simply generated by rerunning the feature extraction on the pseudospectra.

Third, this method eliminates all of the overlapping peaks and excessively small peaks from the library in one step. The peaks that remain are the effective peaks, and only those that might be reasonably detected.

4.4 Spectra Simulation

Ideally, all of the spectra used for this project would be experimentally measured. However, many of the radiation sources needed for this project are not readily available, and certain parts of this project require more spectra than can be measured in a reasonable amount of time. For these reasons, a
Figure 4.7: An $^{152}$Eu spectrum (blue) with the three different libraries overlaid; the height of each stem is proportional to the area of that peak. Top: Complete library, with all 156 peaks. Middle: Kong’s library algorithm. Note that overlapping peaks are removed, but many unresolvable peaks remain. Bottom: new library approach, which eliminates all but one of the small peaks.

Python code was developed to simulate unshielded spectra for any isotope. This code works as follows:

- Load raw library for isotope(s) of interest.
- Load MCNP-generated response functions.
- Convolve response functions with library to get ideal spectra.
• (optional) Sample the ideal spectrum to simulate realistic counting statistics.

The development of this project required MCNP simulations, a simple convolution code, and a sampling code, each of which are described in detail in the following subsections. Complete demonstrations of the end result are shown in Section 4.4.4.

4.4.1 MCNP Simulations

To simulate complete spectra, the detector’s response to a monoenergetic gamma-ray source must be known. This was found via Monte Carlo simulation using MCNP6 (Monte Carlo N-Particle code) [80].

First, a simple geometric model of an Ortec 905-3 [81] was built in MCNP6. This model approximates the detector as a cylindrical shell of aluminum with an inner NaI crystal. The dimensions and materials are approximate; quantities such as the spacing between the crystal and the outer shell are not known. This does have an effect on the detector response, as discussed in Chapter 6, but the effect is small enough to be ignored for current purposes.

Next, the radiation source is defined. For each of these simulations, a monoenergetic gamma-ray source emitting isotropically is placed 20 cm from the detector. Separate simulations are conducted for each energy, ranging from 5 keV to 3000 keV in 5 keV increments.

The simulations were run for $10^8$ particles, which was found to be enough to produce reasonably smooth response functions. These simulations recorded a pulse-height distribution tally over the NaI crystal; for each gamma-ray that is emitted, this tally records the energy that is deposited into the detector [82].

Gaussian energy broadening (GEB) [80] must be used in these simulations to produce appropriately wide photopeaks; the simulations do not include all of the processes within the detector that lead to wider Gaussian peaks, but MCNP can approximate this with the GEB option. The result of one simulation is then the detector response to a monoenergetic source.

Finally, the detector responses must be scaled. By default, MCNP normalizes the tally results such that they are a probability measure. Suppose that a bin of this tally has the value 0.0035; this means that if a gamma-ray
Figure 4.8: An MCNP-generated response to a 1000 keV photon source. There is virtually no back-scatter peak in these detector responses; including a model of the room would remedy this, but for the library generation application this will be used for the backscatter peak will only add unnecessary complexity.

is emitted by the defined source, there is a 0.35% chance that a count would be recorded in this energy bin. However, for reasons that will be more clear in the next subsection, the detector response needs to be scaled such that the photopeak has total area equal to one. This is achieved by using a simple fitting function over the peak region for each detector response function, and dividing the response function by this fitted area.

To produce the detector response at any energy, a simple linear interpolation is used. Let $R(E_i, E)$ be the scaled detector response to a monoenergetic gamma-ray source of energy $E_i$, evaluated at the energy bin $E$. For an arbitrary energy $\hat{E} \in (5, 3000) \text{ keV}$ and $E$ not a multiple of 5 keV (that is, $E$ is not one of the exact energies for which the $R(E_i)$ was simulated), let $j$ be the smallest value such that $E_j > E$. Then the interpolated and scaled detector response at energy $E$ is:

$$\bar{R}(\hat{E}, E) = \frac{\hat{E} - E_{j-1}}{E_j - E_{j-1}} R(E_{j-1}, E) + \frac{E_j - \hat{E}}{E_j - E_{j-1}} R(E_j, E)$$  \hspace{1cm} (4.20)

This interpolation is not perfect. Consider the Gaussian photopeak in
each of these response functions. The interpolated response function then has a weighted average of Gaussian peaks, which is approximately a slightly wider Gaussian. However, the detector response functions were computed with energy steps significantly smaller than the detector resolution, so the approximation that this average peak is Gaussian introduces very little error. For example, at 662 keV, the maximum difference between this interpolated detector response function and a directly simulated one at this energy is 1.1%, and the interpolated peak is 0.4% wider than the expected peak. Therefore, it was concluded that this interpolation error is negligible for this detector, although the energy grid would need to be made finer for a higher resolution detector.

4.4.2 Convolution

Virtually all gamma radiation sources emit gamma-rays with more than one energy, so the response to a monoenergetic source is not enough. To create a pseudospectrum, the detector response function must be convolved with the library for the isotope(s) to be simulated. Consider an isotope with $m$ photopeaks, with peak energies $\{E_j\}_{j=1}^m$ and corresponding branching ratios $\{BR_j\}_{j=1}^m$. Let the detector efficiency as a function of energy be $\epsilon(E)$. Then the smooth pseudospectrum $P$ can be constructed as:

$$
\Psi(E) = \sum_{j=1}^{m} \bar{R}(\bar{E}_j, E) \cdot BR_j \cdot \epsilon(\bar{E}_j)
$$

(4.21)

The result of this is a nearly perfectly smooth pseudospectrum, as demonstrated in Figure 4.8. There are still a few differences between this pseudospectrum and a real one.

First, this spectrum has no background radiation sources. This is intentional, as this formulation allows for the addition of any arbitrary background instead of assuming a fixed background. This should allow for this simple spectrum simulator to be a bit more widely useful.

Second, this method does not include sum peak effects. The particle transport simulations in MCNP are performed sequentially, and the current implementation does not allow for the possibility of sum peak effects. If desired, these could be added in post-processing.
Third, these calculations do not include any shielding material, and adding shielding would require modifying the MCNP geometry and rerunning all of the simulations. This and other future directions for this code are discussed in Chapter 7. This is not an issue for the identification algorithm presented in this dissertation, but it would be necessary to expand these capabilities for other related projects (e.g. [66]).

Finally, the pseudospectra produced by this method are (nearly) perfectly smooth. Physically, this corresponds to an extremely long measurement time, which is not useful for handheld detection scenario modeling. However, this can be remedied by the method in the following subsection.

4.4.3 Sampling

The smooth pseudospectra can be downsampled to mimic the effects of (relatively) short measurement times. The current implementation requires the user to specify the gross number of counts for the spectrum, and simulates each count individually. While this is not the most computationally efficient way this could be implemented (especially for high gross counts), this method is simple to implement and quick for low to medium gross counts.

First, $\Psi(E)$ must be normalized (total area equal to one); this will allow it to be treated as a probability mass function $PMF(E)$. The cumulative distribution function is then computed as:

$$PMF(E_i) = \frac{\Psi(E_i)}{\sum_{i=1}^{n} \Psi(E_i)} \quad (4.22)$$

$$CDF(E_0) = 0 \quad (4.23)$$

$$CDF(E_i) = CDF(E_{i-1}) + PMF(E_i) \quad (4.24)$$

Then, the pseudospectrum can be sampled by generating a uniform random number $x$ on the interval $[0, 1]$, and inverting the CDF to find $E$ such that $CDF(E) = x$. This is then repeated until the desired number of counts has been reached.

Downsampled pseudospectra for $^{152}$Eu are shown in Figure 4.9. As the number of counts increases, the shape of the pseudospectrum approaches the perfectly smooth pseudospectrum, as would be expected from counting
4.4.4 Applications and Extensions

This simulation code has several applications. First, the intended purpose of this project was to simulate spectra for identification algorithm testing, although it has been replaced for this purpose by the Department of Homeland Security (DHS) Algorithm Improvement Program (AIP) code, described later in Chapter 6.

Second, these simulations are used for library generation as described previously, and for some empirical statistics for likelihood model construction. This is described in detail in Section 5.4.6.

There are other applications for this project outside of the scope of this dissertation, such as generating training data for machine learning applications [66], or for generating templates for a template matching algorithm.
CHAPTER 5

STATISTICAL APPROACH TO ISOTOPE IDENTIFICATION

5.1 Introduction

In Chapter 3, various methods for isotope identification were discussed. For this dissertation, a naive Bayesian classifier approach to perform isotope identification has been developed. While this method will be similar in some ways to the library comparison category of algorithms, it is ultimately much more powerful and can leverage much more information to determine radionuclides.

From a raw spectrum (counts vs energy channel), features are extracted using the wavelet/non-negative least squares (NNLS) approach described in Chapter 4. This yields a set of peak energies, areas, and area uncertainties with which identification can be performed. This data is passed to the naive Bayesian classifier, which scores the various spectral features and computes a posterior probability score for each isotope in the library. This approach is highly modular and could easily be modified to include additional data. For example, some of these handheld identifiers also have a neutron detector onboard, which could be used in the identification process. The methodology for Bayesian classifiers is discussed in Section 5.2, and the specific likelihood models are developed in Sections 5.4. The isotope libraries were discussed previously in Chapter 4; some example identifications will be presented in Section 6.1, and full benchmarking is shown in Chapter 6.

5.2 Bayesian Classifiers

From modern medicine to satellite deployment, decisions often need to be made based on imperfect data. One approach is to use a Bayesian classifier, which leverages prior information against observational data to output a
posterior probability distribution for all possible decisions. Given a set of data $D$ used for classification, a set of $m$ different decisions (models) $\{M_i\}_{i=1}^{m}$ and a posterior distribution for that data $P(M|D)$, the Bayesian decision (also called the maximum a posteriori estimator or MAP estimator) is to choose model $M$ that maximizes the posterior (i.e. the most likely model).

$$\hat{\theta}_{Bayes} = \arg \max_{i=1...n} P(M_i|D)$$ (5.1)

For the purpose of isotope identification, each model $M_i$ is an isotope or a specific combination of isotopes.

From Bayes’ Theorem, the posterior can be computed by

$$P(M_i|D) = \frac{P(D|M_i)\pi(M_i)}{f(D)}$$ (5.2)

where $\pi(M_i)$ is the prior probability for model $M_i$ (how likely is model $M_i$ before receiving the data), $P(D|M_i)$ is the likelihood of the data given model $M_i$, and $f(D)$ is the probability of obtaining the data across all possible models. For simple decision making, $f(D)$ is actually unimportant; the maximization in Equation 5.1 is done over the models, so $f(D)$ is just a normalization constant. As demonstrated later, there is further analysis that can be done with the properly normalized data, and computing this term is a simple summation for this specific application. It is assumed that the models used form a complete probability space; this means that the probability of all models must sum to one. In other words, it is assumed to be impossible for the correct identification to be something outside of the library. While this may be a poor assumption in general, it is currently a necessary assumption for this classifier, and it is an assumption that many isotope identifiers must make. In the future, it may be possible to extend this methodology to give a probability that the correct model is none of the models in the library.

Since the sum of the probabilities of all possible models must be equal to one (by completeness of the probability space), we have that:
\[ 1 = \sum_{i=1}^{m} P(M_i|D) = \sum_{i=1}^{m} \frac{P(D|M_i)\pi(M_i)}{f(D)} \]

\[ 1 = \frac{1}{f(D)} \sum_{i=1}^{m} P(D|M_i)\pi(M_i) \]

\[ f(D) = \sum_{i=1}^{m} P(D|M_i)\pi(M_i) \quad (5.3) \]

The choice of prior function \( \pi \) is subjective. It could be used to optimize detector performance for certain scenarios, such as a shipping port, by weighting isotopes based on their actual observed frequency. A detector could even be deployed that learns its prior distribution as it was used more and more. More viable priors for future developments are discussed in Chapter 7.1.

However, for this dissertation a simple prior is used that assigns a constant (uniform) probability to all single-isotope sources, a lower probability for two-isotope sources, a further reduced for three-isotope sources, and so on. This has the effect of punishing models with too many isotopes that may overfit the observed data set.

It’s worth noting that the Bayesian decision is equivalent to the maximum-likelihood (ML) decision if a completely non-informative prior is used. If the prior is non-informative (uniform), it is a constant with respect to changing models, and the posterior distribution only depends on the model in the likelihood term.

\[
\hat{\theta}_{Bayes} = \arg\max_{i=1...n} P(M_i|D) \\
= \arg\max_{i=1...n} \frac{P(D|M_i)\pi(M_i)}{f(D)} \\
= \arg\max_{i=1...n} P(D|M_i) = \hat{\theta}_{ML}
\]

With the currently implemented prior, the Bayesian decision is similar to the maximum-likelihood decision, but the two are not equivalent.

The last piece of Bayes’ theorem is the likelihood function \( P(D|M_i) \). This is the probability that model \( M_i \) would produce this specific data set \( D \). To compute this quantity, the data is broken down into four feature sets and a probability score is defined on each.
\[ P(D|M_i) \approx f_{LR} f_{DR} f_{PP} f_{AR} \] (5.4)

Each of these terms were constructed to mimic the thought process of a trained spectroscopist performing manual peak-based isotope identification. These models are defined and developed in the following sections; in brief, they are:

- \( f_{LR} \): the library representation score, accounts for how fully the library is represented by the data
- \( f_{DR} \): the data representation score, accounts for how fully the data is represented by the library
- \( f_{PP} \): the peak position score, scores how closely the peak positions in the data match the library peak positions
- \( f_{AR} \): the area ratio score, accounts for the peak areas observed in the data set

By breaking the likelihood into multiple probability scores, independence of these feature sets is inherently assumed. In truth, these features are not independent. This false assumption is the difference between a true Bayesian classifier and a naive Bayesian classifier. This simplification reduces the complexity of the models and the computations (no need to account for correlations between feature sets). Further, there are some cases where a naive Bayesian classifier competes with or even out-performs its non-naive counterparts for classification, e.g. [83]. In the future, if adequate performance is not accomplished by this naive classifier, a non-naive or semi-naive [84] Bayesian classifier may be studied for this problem. Each of the aforementioned likelihood components is developed in the following sections, and other possible likelihood components are discussed in Section 5.4.8.

5.3 Peak Pairing

Before computing the likelihood, each model in the library is compared to the data to determine which library peaks correspond with each data peak. For each isotope in the library, a simple nearest neighbor test is used to find
the library peak that corresponds to each data peak. Library peaks that are too distant in energy from the data peaks are not considered matches, even if they are the closest corresponding peak. Consider the simple case in Figure 5.1. Both $^{137}$Cs and $^{54}$Mn have one large photopeak, but with the given energy calibration of the spectrum it would be unreasonable to assume that the photopeak belongs to $^{54}$Mn.

To determine what is “too distant in energy” to be considered a match, a limit was set with a neighborhood window that has simple linear energy dependence. A library peak at energy $E_j$ is considered a match to the data peak at energy $e_k$ if both of the following conditions are met:

$$E_j = \text{argmin} |E_j - e_k|$$

$$|E_j - e_k| \leq w(E_j) := \theta_{w}^{\min} + \frac{E_D}{3000} \times (\theta_{w}^{\max} - \theta_{w}^{\min})$$

where $\theta_{w}^{\min}$ and $\theta_{w}^{\max}$ are the minimum and maximum neighborhood limits respectively. This neighborhood limit varies so that the effects of poor calibrations can be mitigated appropriately. A poor calibration will have a larger total effect on high energy peaks than lower energy peaks (in terms of total keV of energy shift, not as percentage necessarily). Because peaks are narrower at lower energies (peak FWHM increases as $E^{1/2}$ [10]) and isotopes tend to have more peaks at lower energies (e.g. $^{152}$Eu has 73 peaks below 500 keV, 56 peaks in [500, 1000) keV, 21 peaks in [1000, 1500), and 6 peaks above

Figure 5.1: Top: $^{137}$Cs spectrum with its library overlaid. Bottom: $^{137}$Cs spectrum with the library peak of $^{54}$Mn overlaid.
1500 keV), it is also necessary to have a smaller allowable neighborhood at those low energies to prevent incorrect peak matching. The values for these parameters are found later in the optimization stage; reasonable estimates for $w_{\text{min}}$ and $w_{\text{max}}$ are 15 keV and 80 keV respectively.

There are several ways this pairing process could be improved further. Imagine that the spectrum was calibrated poorly (an unfortunately common scenario in field usage of these handheld detectors). If the calibration is poor enough, peaks could be mismatched at this step, which will hurt several of the likelihood scoring functions later. Instead of using the poor calibration, the spectrum can be recalibrated on the fly. Depending on the situation, the recalibration could be performed in a variety of ways. If a high enough background is present, the calibration could be adjusted to improve the positions of the 1460 keV peak of $^{40}$K and 2614 keV peak from the $^{232}$Th decay chain, as well as others. However, this method may not always be applicable, and isn’t used here.

Instead, a new calibration can be chosen during the model scoring step for each model individually. Instead of trying to force a calibration based on background, assume that the model $M_i$ being considered is the correct source; this is actually not a problem, as this is being used for the calculation of the likelihood $P(D|M_i)$, the probability of obtaining data set $D$ given that model $M_i$ is correct. The updated calibration will be chosen to maximize the peak position score, $f_{\text{PP}}$, the first component of the likelihood model, which scores how well the data peaks match the library peaks in energy. The exact re-calibration used is described after the discussion of $f_{\text{PP}}$ in Section 5.4.3.

After this matching is done for each observed peak, the computations for the likelihood function can commence.

5.4 Likelihood Models

Using the set of matched peaks, the likelihood is calculated as the product of the four scoring functions. These likelihood components are designed to replicate the way that a trained spectroscopist looks at a spectrum and makes identifications. By mimicking this logic, we aim to achieve performance comparable to a spectroscopist’s manual identifications, which would be a significant improvement for automated identifiers [7].
Figure 5.2: Motivation for $f_{LR}$. Data lines represent peaks extracted from a spectrum, while library lines correspond to the expected peaks in a given library model. Each of the data peaks clearly is strongly associated with a library peak, and we could say the data is well-represented by the library. However, the converse is not true; many library peaks are not represented in the data set.

5.4.1 $f_{LR}$: Library Representation Score

Generally, when a spectroscopist is given a spectrum and asked to identify the source, they will look at the peaks in a spectrum. Each isotope has its own set of peak energies and branching ratios, and by comparing the peaks in the data against a library of isotopes, an accurate identification can generally be made. To decide if a particular model (isotope or combination of isotopes) is responsible for the spectrum, the spectroscopist will first decide if the library model is well-represented by the data. Are the peaks that should be observed for this library model present in the spectrum? If not, is it reasonable to assume that shielding would have eliminated the missing peaks? This logic is captured by the library representation term, $f_{LR}$. Two different possibilities for this function are discussed here.

Consider the hypothetical data set and library model shown in Figure 5.2. In this example, the data peaks are all explained by this library model, but the converse is not true. Many of this library model’s peaks were not observed in the data set, so this model should be given a low score despite explaining the observation.

Suppose that the library isotope $M$ contains $n$ peaks, with areas $A_1$, $A_2$, $\ldots$, $A_n$, and let $\delta_j$ be the indicator function that the $j^{th}$ library peak was
matched (i.e. $\delta_i = 1$ if the $i^{th}$ peak was matched, 0 otherwise). Then $f_{LR}$ is:

$$f_{LR}(M, D) = \prod_{j=1}^{n} \left( 1 - \theta_{LR} \frac{A_j}{\max A_j} (1 - \delta_j) \right)$$

(5.5)

where $\theta_{LR}$ is a weight parameter. This function may be more easily understood as follows. A score is given to each peak in the library. If the library peak was matched to a data peak, it is given a score of 1. If not, it is given a score of $\theta_{LR}A_j/\max\{A_j\}$. The final score $f_{LR}$ is then the product of each peak score.

This form is easily understood, but is missing an important logical step. Suppose that the data was produced by a model that contains multiple photopeaks, such as the 1173 keV and 1332 keV peaks of $^{60}\text{Co}$. If a lower energy peak is observed, the higher energy peaks should be observed as well. No shielding configuration will eliminate high energy peaks before low energy peaks. For this reason, if a peak was observed at 1173 keV but not at 1332 keV, $^{60}\text{Co}$ should not be a candidate for identification. On the other hand, it is reasonable to only detect the higher energy peaks but not the lower energy peaks if some attenuating materials are present.

For these reasons, the library representation score was modified. For notational convenience, define the set of matched peak indices $s_m$ and the set of unmatched peak indices $s_u$:

$$s_m = \{j| j^{th} \text{ library peak was matched}\}$$

(5.6)

$$s_u = \{j| j^{th} \text{ library peak was not matched}\}$$

(5.7)

$$k = \min s_m$$

(5.8)

Then the modified library representation score is:

$$f_{LR} = \prod_{j \in s_m^{1}} \left( 1 - \theta_{LR1} \frac{A_j}{\max\{A_{m}\}_{m \in s_m}, A_j} \right) \cdot \prod_{j \in s_u, j < k} \left( 1 - \theta_{LR2} \frac{A_j}{\max\{A_{m}\}_{m \in s_m}, A_j} \right) \cdot \prod_{j \in s_u, j > k} \left( 1 - \theta_{LR2} \frac{A_j}{\max\{A_{m}\}_{m \in s_m}, A_j} \right)$$

(5.9)
Figure 5.3: Motivation for $f_{DR}$. Data lines represent peaks extracted from a spectrum, while library lines correspond to the expected peaks in a given library model. Each peak in the library corresponds strongly to a data peak, but many of the data peaks are not represented by this library model.

This score is similar to the original model, but with a few key differences. First, peaks at a lower energy than the lowest matched peak are not penalized as harshly; this is to capture the logic described above. However, if a lower energy peak was matched, then a higher energy peak should be penalized more harshly if it is not paired.

5.4.2 $f_{DR}$: Data Representation Score

After considering how well the library was represented, the spectroscopist will then consider if the data was adequately explained by this model. The previous score roughly checks to see if the library model could be present, but it does not evaluate how much of the data set was explained by the model. In Figure 5.3, an example is shown where all of the library model peaks do correspond to data peaks (so $f_{LR}$ would be 1), but many data peaks are not accounted for. Thus, the library representation score is not sufficient for isotope identification, as the data could easily contain all of the model’s peaks and more.

If the data set $D$ contains $d$ peaks with areas $\bar{A}_1$, $\bar{A}_2$, \ldots $\bar{A}_d$, and $\delta_k$ is in the indicator that the $k^{th}$ data peak was matched to a library peak, then $f_{DR}$ is:
\[ f_{DR}(M, D) = \prod_{k=1}^{d} \left( 1 - \theta_{DR} \frac{\bar{A}_k}{\max \bar{A}_k} (1 - \bar{\delta}_k) \right). \] (5.10)

By the same line of reasoning as in the library representation score, \( f_{DR} \) can be modified to better capture the thought process of a spectroscopist.

Again, define the set of matched peak indices \( \bar{s}_m \) and the set of unmatched peak indices \( \bar{s}_u \):

\[ \bar{s}_m = \{ j \mid j^{th} \text{ data peak was matched} \} \] (5.11)
\[ \bar{s}_u = \{ j \mid j^{th} \text{ data peak was not matched} \} \] (5.12)
\[ k = \min s_m \] (5.13)

Then the modified library representation score is:

\[
\begin{align*}
    f_{DR} &= \prod_{j \in \bar{s}_m} 1 \\
    &\prod_{j \in \bar{s}_u, j < k} \left( 1 - \theta_{DR1} \frac{\bar{A}_j}{\max (\{ A_m \}_{m \in \bar{s}_m}, A_j)} \right) \\
    &\prod_{j \in \bar{s}_u, j > k} \left( 1 - \theta_{DR2} \frac{\bar{A}_j}{\max (\{ A_m \}_{m \in \bar{s}_m}, A_j)} \right)
\end{align*}
\] (5.14)

The construction of this score is precisely analogous to the library representation score, but the values for these parameters will not be the same. This is due to imperfections in the feature extraction algorithm (e.g. there is not a 100% chance of detecting every peak) and in the detection scenarios.

The library representation and data representation scores together can produce accurate isotope identifications for many simple cases, but there are many other cases that they are unable to handle. They don’t consider how closely in energy the data peaks matched the library peaks, nor do they actually account for the area of the observed peaks. This leads to the inclusion of the next two model terms.
5.4.3 $f_{PP}$: Peak Position Score

The previous scoring functions evaluated if each of the library and data peaks were paired, but they did not address whether each of the pairings was actually a good fit in energy and area. With a perfect calibration and a perfect feature extraction algorithm, the data peaks would exactly align in energy with the correct library model’s peaks. However, there are several factors that cause the observed data peak centroids to fluctuate. These will lead to the inclusion of the third likelihood term, $f_{PP}$.

First, imperfect calibrations will cause the spectrum to shift in energy. This could be caused by temperature fluctuations, as NaI is sensitive to even small changes in room temperature. The imperfect calibration could also be due to user error, either in the calibration stage or by waiting too long to recalibrate the detector. Second, poor counting statistics will increase the error in the peak fitting process, increasing the variation further. Finally, the feature extraction method itself may bias the peak centroids.

To account for these effects, the peak position is given a Gaussian score. Assume that, on average, the data peak centroid should be the same as the library centroid. Since the fits are done with non-negative least squares (NNLS), the estimation of the peak centroid of a Gaussian peak should be unbiased. Finally, because each estimate of the centroid is essentially an average, by the Central Limit Theorem the fit centroid distribution should be Gaussian as well.

To further justify this score, the distributions can be observed empirically by taking many spectra under the same conditions and fitting all of the peaks. The empirical distributions for the 1173 keV and 1332 keV peaks of $^{60}$Co are shown in Figure 5.4. Interestingly, the distributions for each of the peaks seem to be skewed in opposite directions somewhat; this effect will be further explored in the future. The true overall distribution is an average over all possible scenarios, but since it is not feasible to observe all scenarios, a different approach much be used to determine the function used for this computation.

In each of the possible scenarios, the distribution is approximately Gaussian, and all should have the same expected value (centroid). Averaging over all possible scenarios would then still yield a Gaussian distribution. Suppose $X_1, X_2, \ldots$ are all Gaussian random variables with the same expectation, $\mu$,
Figure 5.4: Empirical peak position distributions for $^{60}$Co. Top: 1173 keV peak. Bottom: 1332 keV peak. These were generated with 2000 15-second spectra of a 0.25 $\mu$Ci $^{60}$Co source. The true distributions will be broader, as the 6.6 hours necessary for this measurement will not adequately sample temperature flucuations or all shielding configurations.

but with different variances, $\sigma_i^2$. Then any average of these random variables, $S = a_1X_1 + a_2X_2 + ...$ is also a Gaussian random variable with expectation $\mu$ and variance $\sigma_S^2 = a_1^2\sigma_1^2 + a_2^2\sigma_2^2 + ...$. Thus, the true empirical distribution for the data peak energy $E_D$ is Gaussian, with expectation equal to the library energy $E_L$ and an unknown variance $\sigma(E_L)^2$. This variance can be treated as another optimization parameter for the Bayesian classifier.

The peak position term $f_{PP}$ is computed as follows. For the library energy $E_L$, suppose the nearest data peak has energy $E_D$. The peak neighborhood, as discussed in Section 5.3, is $w(E_L)$. Then the peak position score $f_{PP}$ for this library peak is:

$$f_{PP}(E_D, E_L) = \begin{cases} 
\exp \left( -\frac{(E_D - E_L)^2}{2\sigma^2} \right) & \text{if } |E_D - E_L| \leq w(E_L) \\
\theta_{PP} & \text{if } |E_D - E_L| > w(E_L)
\end{cases}$$
and the total peak position score for model $M_i$ is the product of each individual score. An example of this score for the 1332 keV peak of $^{60}\text{Co}$ is shown in Figure 5.5. Above, $\theta_{PP}$ is the penalty applied to unmatched peaks. While these peaks were penalized by earlier model terms, they must also be reduced here. Otherwise, matched peaks would receive some score less than one, reducing the final posterior, while unmatched peaks here would effectively get a score of one, which is relatively increasing the posterior. The penalty $\theta_{PP}$ is then for consistency. Like the neighborhood term $w(E_L)$, $\theta_{PP}$ can be a function of energy. Because higher energy peaks are less attenuated by intermediate materials, they are less likely to be unmatched compared to low energy peaks. Thus, setting $\theta_{PP}$ to be a decreasing function of energy is appropriate, and a simple linear function is used.

### 5.4.4 Using $f_{PP}$ for Dynamic Recalibrations

In the peak matching section, it was suggested that a recalibration could be performed on the fly. A recalibration has been implemented that seeks to maximize this peak position $f_{PP}$ score. Two different versions have been derived and tested.

The first of these applies a fixed percentage adjustment to the data peak energies (a scale adjustment). If the original data peak energies are $E_i$, let the new peak energies have the form $s\hat{E}_i$, where $s$ is chosen to maximize this
likelihood score. The peak position score becomes:

$$f_{PP}(\bar{E}, E, s) = \prod_{i=1}^{n} \exp\left(-\frac{(s\bar{E}_i - E_i)^2}{2\sigma_i^2}\right)$$  \hspace{1cm} (5.15)

The optimal scale parameter $s$ is found with straightforward calculus.

$$\left.\frac{\partial f_{PP}(\bar{E}, E, S)}{\partial S}\right|_{S=s} = 0$$ \hspace{1cm} (5.16)

$$s = \frac{\sum_{i=1}^{n} \frac{E_i E_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{E_i^2}{\sigma_i^2}}$$ \hspace{1cm} (5.17)

A more flexible recalibration would be to introduce a shift and a scale parameter, such that the new data peak energy is $s\bar{E}_i + b$. Again, the optimal solution is a straightforward derivation.

$$f_{PP}(\bar{E}, E, s, b) = \arg \max_s \left( \exp \sum_{i=1}^{n} \left(-\frac{(sE_i + b - E_i)^2}{2\sigma_i^2}\right) \right)$$ \hspace{1cm} (5.18)

$$\left.\frac{\partial f_{PP}(\bar{E}, E, S, B)}{\partial S}\right|_{S=s} = 0$$ \hspace{1cm} (5.19)

$$\left.\frac{\partial f_{PP}(\bar{E}, E, S, B)}{\partial B}\right|_{B=b} = 0$$ \hspace{1cm} (5.20)

$$s = \frac{\sum_{i=1}^{n} \frac{E_i E_i}{\sigma_i^2} \sum_{i=1}^{n} \frac{1}{\sigma_i^2} - \sum_{i=1}^{n} \frac{E_i^2}{\sigma_i^2} \sum_{i=1}^{n} \frac{E_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{E_i^2}{\sigma_i^2} \sum_{i=1}^{n} \frac{1}{\sigma_i^2} - (\sum_{i=1}^{n} \frac{E_i}{\sigma_i^2})^2}$$ \hspace{1cm} (5.21)

$$b = \frac{\sum_{i=1}^{n} \frac{E_i}{\sigma_i^2} - s \sum_{i=1}^{n} \frac{E_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}}$$ \hspace{1cm} (5.22)

This two parameter recalibration creates more accurate peak recalibrations, but unfortunately in testing it tends to overfit the data for many of the wrong identifications. For this reason, the simpler recalibration is used in the current algorithm. If a higher resolution detector was used, more peaks would generally be identified, and the two parameter calibration would likely perform better.
5.4.5 \(f_{AR}: \) Without Area Uncertainties

The last major feature set for a spectroscopist to consider is the areas of the observed peaks. There are many instances where looking at peak energies is not enough for a confident measurement, such as the case of \(^{239}\text{Pu}\) (major peaks at 378.5 and 413.7 keV) and \(^{177}\text{m}\text{Lu}\) (major peaks at 375.0 and 413.7 keV). However, the branching ratios of these peaks differ enough to make accurate identification possible.

To use area information, a spectroscopist can only compare peak areas relative to other areas because changing the count time, source activity, or intermediate materials all affect the raw number of counts observed. The current iteration of the Bayesian classifier uses the area ratios of neighboring peaks to improve estimates of the likelihood function. Using neighboring peaks reduces the effects of shielding, as discussed below.

Two different versions of this scoring function have been developed. The original version is presented in this subsection and only uses the area ratio information. An alternative function has been built that incorporates the area uncertainty information, and is presented in Section 5.4.6.

Suppose the data contains two peaks (e.g. the two photopeaks of \(^{60}\text{Co}\), as shown in Figure 5.6), with energies \(E_1\) and \(E_2\) and areas \(A_1\) and \(A_2\), respectively, while model \(M_i\) contains peaks \(E_1\) and \(E_2\) with areas \(A_1\) and \(A_2\) that were matched to the data peaks. The data area ratio \(\bar{r} = \frac{A_1}{A_2}\) is then compared against the model area ratio \(R = \frac{A_1}{A_2}\).

When \(E_1 \approx E_2\), we have \(\mu(E_1) \approx \mu(E_2)\) and the observed ratio approaches the expected library ratio. Also, a larger difference between the energies of these peaks will result in a larger uncertainty in the area ratio because the difference between the attenuation coefficients is larger and shielding can affect the area ratios more.

For a completely ideal spectrum, the data area ratio \(\bar{r}\) would be exactly equal to the model ratio \(R\). Introducing shielding materials will reduce the data ratio \(\bar{r}\) because the interaction cross-section \(\mu(E)\) of the intermediate material(s) decreases with energy:

\[
\bar{r} = \frac{A_1 \exp(-\mu(E_1)x)}{A_2 \exp(-\mu(E_2)x)} = R \cdot \exp(\mu(E_2)x - \mu(E_1)x)
\]

The area ratio score should then not penalize ratios that are reasonably smaller than the library ratio. When these peaks are close together in energy,
the cross-sections should be similar, and the reduction in area ratio is less significant. However, intermediate materials should not ever increase this ratio, so a larger than expected area ratio should be penalized.

In reality, imperfect counting statistics and realistic feature extraction introduce more error into the area calculation, which can increase or decrease the ratio $\bar{r}$.

Summarizing, the area ratio score should obey the following logic:

1. $\bar{r} \approx R$: no penalty (score of one).
2. $\bar{r} < R$: small penalty, as the ratio will decrease as more shielding is added. Penalty should increase as $\bar{r}$ decreases.
3. $\bar{r} > R$: outside of a small margin for error in the area calculations, this should be penalized heavily and should increase exponentially with increasing $\bar{r}$.

Define the left and right bounds $B_l$ and $B_r$ for the region of no penalty as follows:

$$B_l = R \cdot \exp \left( -\frac{E_2 - E_1}{3000} \right)$$

$$B_r = 1.1 \cdot R$$

The lower bound decreases as the energy difference between the peaks increases to allow for a greater effect from attenuating material. The area ratio score is then estimated by the preceding logic and is given by:
Figure 5.7: $f_{AR}$ for the $^{60}$Co photopeaks at 1173 and 1332 keV. These peaks have a library area ratio of 1.00

$$f_{AR}(\bar{r}, R) = \begin{cases} 
1 - e^{-\bar{r}} + e^{-B_l} & \text{if } r \leq B_l \\
1 & \text{if } B_l < \bar{r} \leq B_r \\
e^{B_r-r} & \text{if } B_r < \bar{r}
\end{cases} \quad (5.23)$$

A plot of this function is shown in Figure 5.7 for the two peaks of $^{60}$Co, which have a library ratio of approximately one.

This area ratio score is computed for each neighboring pair of peaks in the data, and the final area ratio score is the product of all of these.

5.4.6 $f_{AR}$: Incorporating Area Uncertainties

As discussed in Section 4.2, an extension to the wavelet/NNLS algorithm to compute the variances of the peak area fit has been developed [75, 76]. Conceptually, if a peak has a large uncertainty, its area should not be scored as harshly. A likelihood function for this has been built empirically via spectra simulation and kernel density estimation.

First, the area uncertainty calculation must be understood. Rather than use the standard deviation on the area fitting, a scaled standard deviation is used throughout this work, and this will be generally referred to as the area uncertainty. The uncertainty $u_i$ associated with the $i^{th}$ peak is:
Figure 5.8: Area uncertainty for single peaks as a function of expected peak area. This was produced by generating spectra with the simulation code and extracting peak information with the wavelet/NNLS code. While the area uncertainty relative to the area increases dramatically with decreasing area, peaks below 100 counts are generally difficult to detect at all, let alone fit accurately. Top: Uncertainty versus area. Bottom: Relative uncertainty \( \frac{u_i}{A_i} \) vs area.

\[ u_i = \frac{\sqrt{\text{diag}(C_k)_j}}{A_i} \]  \hspace{1cm} (5.24)

where \( C_k \) is the covariance matrix described in Section 4.2, \( j \) is the index of the channel corresponding to the photopeak, and \( A_i \) is the area of the peak.

As shown in Figure 5.8, the uncertainty is strongly a function of the peak area, which is expected to an extent. When a peak has low area, each channel has a large amount of Poissonian variation, making the peak fitting difficult.

As in the previous model for \( f_{AR} \), the area ratio score will be done pairwise on neighboring peaks. Let \( A_1 \) and \( A_2 \) be the peak areas found with the wavelet/NNLS algorithm, \( u_1 \) and \( u_2 \) be the corresponding area uncertainties. Then the area ratio \( r = \frac{A_1}{A_2} \) is scored against the expected area ratio \( R \). In an ideal spectrum with perfect counting statistics and no attenuation, the observed ratio \( r \) should be the same as the expected \( R \). In reality, interme-

75
diate materials will reduce this ratio, the significance of which depends on the materials and the peak energies. Poor signal-to-noise ratio (SNR) can dramatically increase or decrease the observed $R$, depending on the areas of each of the peaks.

Ideally, the effect of the SNR will be captured in the peak uncertainties $u_1$ and $u_2$. For small uncertainties, this effect should be minimal, and the observed ratio should be the same (little effect from shielding) as the expected ratio or be somewhat less (more effect from shielding materials). When $u_1$ is large but $u_2$ is small, only some additional deviation in the observed $r$ is reasonable, as varying $A_1$ by 20% changes $r$ by 20%. However, when $u_2$ is large, the observed ratio may be wildly different than the expected ratio, and the peak area information should only contribute weakly to the isotope identification.

To build this scoring function, 200,000 spectra were simulated in MCNP6, using the method previously described in Section 4.4. The simulation geometry assumed a model of an Ortec 905-3 NaI detector [81]. For each simulated spectrum, random energies and branching ratios were sampled from uniform distributions, and a point source with these characteristics was created. Cosmic and terrestrial backgrounds were not included in the simulations. Including these backgrounds would either increase the area uncertainties slightly or, in some cases of source photopeaks overlapping with background peaks, cause peaks to be missed by the wavelet/NNLS.

The wavelet/NNLS algorithm then extracted peak information from each of these simulated spectra. For each spectrum, this consists of a pair of peak energies, areas, and area uncertainties.

Finally, the probability density was estimated using kernel density estimation (KDE). During isotope identification, this expected area ratio $R$ is the ratio of the branching ratios for the two photopeaks, convolved with detector efficiency. However, at this stage, the expected area ratio $R$ is known from the simulation stage.

KDE is used to directly estimate the probability $P(r, u_1, u_2, R)$. However, an isotope should not be penalized for the area uncertainties, but instead should be evaluated on the observed data ratio $r$ as a function of the area uncertainties and the library ratio. Further, it is actually desired to allow $\max f_{AR}(r, u_1, u_2, R) = 1$ instead of being a normalized probability density (integrating to one). If a normalized distribution was used, the maximum
score would be less than one, which would effectively be penalizing every observed ratio, even if it precisely matched the library ratio. This could be performed by including an additional penalty for missed peak pairs, but this approach simplifies the optimization and yields the same net result.

For these reasons, the area ratio score is set to be proportional to the conditional density $P(r|u_1, u_2, R)$:

$$f_{AR}(r, u_1, u_2, R) := \frac{P(r|u_1, u_2, R)}{\max_\rho P(\rho|u_1, u_2, R)}$$

Examples of this scoring function are shown in Figure 5.9. Ultimately, the shape of this function is similar to previous models for the area ratio [85, 86], but the area uncertainties can scale and distort the score.

5.4.7 Computation Summary

For each model $M_i$ in the library, the likelihood $P(D|M_i)$ is approximated as the product of four different components, each developed to mimic the manual identification process of a trained spectroscopist.

- $f_{LR}$: the library representation score
- $f_{DR}$: the data representation score
- $f_{PP}$: the peak position score
After the likelihoods have been calculated for every model, the posterior probability for each model is calculated by convolving the likelihoods with the prior function and renormalizing.

The renormalization makes it possible to examine the results in two different ways. First, the model posteriors give the probability that a given model is correct, meaning that the isotopes in that model are all present and no other isotopes contributed to the data set. This is useful for total identification, but it is also possible to obtain the probability that any individual isotope is present by summing across all models that the isotope is in. This is derived in Section 5.5. This is particularly useful in complicated spectra of multiple sources, where it may be difficult to ascertain the exact combination of isotopes that produced the spectrum with high confidence. Sometimes, the identification of individual isotopes with a high degree of confidence is still possible in this complicated spectra. An example of this will be shown in Section 6.1.

5.4.8 Other Likelihood Models

The presented likelihood models use the peak energies, areas, and uncertainties to make identifications. However, there are other features in the spectrum that can be used to improve the Bayesian classifier’s performance.

First, peak width information could be used, as the expected peak width is a measurable function of detector energy for NaI detectors. If a peak is wider than expected, there are a few possible explanations. First, there could be several photopeaks that are unresolvably close in energy, resulting in a wider than expected photopeak. A wider peak could also be caused by a different process, such as annihilation, single or double-escape, or other nuclear interactions like the $^1\text{H}(n,\gamma)^2\text{H}$ reaction (which causes the 2.2 MeV peak in the BeRP spectrum shown in Figure 2.13). On the other hand, narrower than expected peaks are detected in low-count scenarios (see the small 2614 keV peak in Figure 4.3 from the background $^{232}\text{Th}$ decay chain). The wavelet/NNLS code also sometimes creates peak fits that are narrower than expected; this typically occurs when two resolvable peaks have some amount of overlap, such as the 1173 keV and 1332 keV peaks of $^{60}\text{Co}$ (see
Ideally, peak width information would inform the identification process in a useful way, but the tested implementations did not find this to be the case. There are three problems in its implementation, all of which are at least partially attributable to the wavelet/NNLS code. First, the wavelet/NNLS code filters out peaks if their width is not within a reasonable window of the expected value. Second, the widths the code finds are not always reliable, as demonstrated in the $^{60}$Co example of Figure 4.3. Finally, the NaI detectors are relatively low-resolution; a set of unresolvably-overlapping peaks creates an effective peak, which if detected has a peak width that may not be significantly different than the expected value. However, the peak width may be more informative in medium-resolution detectors, where more peaks will be detectable and more peak width information would be computable.

These peaks from other processes can also be used to improve identifications to an extent. Detection of escape peaks could be used to improve the confidence on higher energy peaks (e.g. the peak 511 keV or 1022 keV above the single or double escape peak). Detection of annihilation peaks or peaks from other nuclear processes can be used to some extent. Detecting the annihilation peak at 511 keV really only tells the spectroscopist that a positron emitter (beta plus decay) may be present, but this could also mean that pair production is occurring. Currently, our identification code will output a warning that a possible annihilation peak was detected.

Similarly, the presence of a peak like the 2.2 MeV peak from the neutron absorption of hydrogen informs the spectroscopist that a neutron source is present. Some of these handheld detectors include a small neutron detector onboard, which will generally make more reliable measurements of neutron flux. The identification code developed for this dissertation will output a note that a neutron source may be present; however, detection of the 2.2 MeV peak is not currently reliable with the wavelet/NNLS code, as this peak is wide enough that the peak detection algorithm generally filters it out.

Next, if Compton edges were reliably detected and distinguished from real peaks, it would be possible to make a better estimate of peak centroids. This feature detection has not been explored by this work, and is likely impossible with the current feature detection algorithm. This would be better suited to spectra with more counts than is typical for these handheld scenarios.
Finally, the shape of the Compton continuum can be used to estimate any attenuating materials between the source and the detector. This type of modeling is included in GADRAS [87], for example, but this is computationally complex and is currently not suited for handheld detectors with limited computational resources.

5.5 Algorithm Output

The immediate output of the isotope identification algorithm is a posterior probability for each model in the library. For clarity, we will refer to this posterior as the *model posterior probability*.

Each model here is an isotope or a combination of isotopes. Each of the likelihood models were constructed so that each of the models would be orthogonal; that is,

\[ P(M_i \text{ and } M_j | D) = 0 \text{ for } i \neq j \]  

(5.26)

This can be understood as model \( M_i \) is the model that assumes its isotopes and only its isotopes are present and no other contributions to the spectrum were made.

This paradigm makes it simple to compute the probability that a given isotope \( iso_j \) is present given a data set. Note that:

\[ P(M_i \text{ and } iso_j | D) = 0 \text{ if } iso_j \notin M_i \]  

(5.27)

\[ P(M_i \text{ and } iso_j | D) = P(M_i | D) \text{ if } iso_j \in M_i \]  

(5.28)

Because all of the model posterior probabilities sum to one (in other words, the models span the probability space), we can find the isotope posterior probability \( P(iso_j | D) \). Let \( \Xi_j \) be the set of all \( i \) such that \( iso_j \in M_i \). Then
\[ \Xi_j : = \{ i \mid iso_j \in M_i \} \]  

(5.29)

\[ P(iso_j|D) = \sum_{i=1}^{n} P(iso_j \text{ and } M_i|D) \]  

(5.30)

\[ = \sum_{i \in \Xi_j} P(iso_j \text{ and } M_i|D) + \sum_{i \notin \Xi_j} P(iso_j \text{ and } M_i|D) \]  

(5.31)

\[ = \sum_{i \in \Xi_j} P(M_i|D) \]  

(5.32)

Thus, by simply summing over all of the model posteriors that contain iso\(_j\), the isotope posterior probability can be found.

This gives the user another way to interpret the results of an identification. It is particularly useful when the maximum model posterior is low, as it may still be able to confidently identify some of the sources in a spectrum. Examples of this are shown in the following chapter.
CHAPTER 6

RESULTS AND BENCHMARKING
ALGORITHM PERFORMANCE

To understand the performance of this algorithm, some identification demonstrations are presented in Section 6.1. Afterwards, performance benchmarks created with the Department of Homeland Security (DHS) Algorithm Improvement Program (AIP) are presented. Finally, a brief experiment and discussion on the limits of detectability is presented in Section 6.5.

6.1 ID Demonstrations

Several demonstrations of this identification algorithm are presented here, ranging from a simple $^{241}\text{Am}$ spectrum to shielded HEU. For all of these demonstrations, the wavelet/NNLS code was used to extract peak information from the spectra. Except for third example ($^{133}\text{Ba}$ and $^{152}\text{Eu}$), none of these test cases used background subtraction.

6.1.1 Demo: Am-241

One of the simplest isotopes to identify is $^{241}\text{Am}$. This isotope has a large photopeak at 59.5 keV, and one small one at 662.5 keV. A 60 second spectrum is shown in Figure 6.1.

The model posteriors for the five most probable models are shown in Table 6.1. The correct identification only has a posterior of 24.4%. There are two reasons for this. First, only the peak at 59.5 keV is being used for this identification; the limited information available leads to an unconfident decision. Second, there are other isotopes in the library that have very few peaks and that have a peak near this energy. Because these other isotopes are also reasonable, the posterior for the correct model is reduced further.

In Table 6.2, the top isotope posterior probabilities are listed. As discussed
Figure 6.1: A 60 second $^{241}$Am spectrum. A large photopeak is observed at 59.5 keV, while a small peak at 662.5 keV is just barely visible. A background peak from $^{40}$K is also detectable.

Table 6.1: Model Posteriors for the Am-241 Spectrum

<table>
<thead>
<tr>
<th>Isotope 1</th>
<th>Isotope 2</th>
<th>Isotope 3</th>
<th>Model Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Am-241</td>
<td></td>
<td></td>
<td>0.2624</td>
</tr>
<tr>
<td>Am-241</td>
<td>Ir-192</td>
<td></td>
<td>0.17493</td>
</tr>
<tr>
<td>Am-241</td>
<td>Tl-201</td>
<td></td>
<td>0.17493</td>
</tr>
<tr>
<td>Ir-192</td>
<td></td>
<td></td>
<td>0.16045</td>
</tr>
<tr>
<td>U-238</td>
<td></td>
<td></td>
<td>0.086553</td>
</tr>
</tbody>
</table>

Table 6.2: Isotope Posteriors for the Am-241 Spectrum

<table>
<thead>
<tr>
<th>Isotope</th>
<th>Presence Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Am-241</td>
<td>0.68907</td>
</tr>
<tr>
<td>Ir-192</td>
<td>0.33892</td>
</tr>
<tr>
<td>Tl-201</td>
<td>0.18714</td>
</tr>
<tr>
<td>U-238</td>
<td>0.10644</td>
</tr>
<tr>
<td>Pu-240</td>
<td>0.028777</td>
</tr>
</tbody>
</table>

in the last chapter, these are obtained by integrating across all models that contain each of these isotopes. Using this information, the detection of $^{241}$Am is more confident overall.
Figure 6.2: A 60 second $^{60}$Co spectrum. Large photopeaks are observed at 1173 and 1332 keV. A background peak from $^{40}$K is also visible.

Table 6.3: Model Posteriors for the Co-60 Spectrum

<table>
<thead>
<tr>
<th>Isotope 1</th>
<th>Isotope 2</th>
<th>Isotope 3</th>
<th>Model Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-60</td>
<td></td>
<td></td>
<td>0.82692</td>
</tr>
<tr>
<td>Co-60</td>
<td>K-40</td>
<td></td>
<td>0.027701</td>
</tr>
<tr>
<td>Co-60</td>
<td>I-125</td>
<td></td>
<td>0.027567</td>
</tr>
<tr>
<td>Am-241</td>
<td>Co-60</td>
<td></td>
<td>0.001318</td>
</tr>
<tr>
<td>Co-60</td>
<td>Cs-137</td>
<td></td>
<td>0.001308</td>
</tr>
</tbody>
</table>

6.1.2 Demo: Co-60

Next, another simple demonstration is shown. $^{60}$Co has two significant photopeaks at 1173 and 1332 keV, as shown in Figure 6.2. The model posteriors from the Bayesian classifier are presented in Table 6.3, while the isotope posteriors are shown in Table 6.4. In this example, there are more peak data that is used in the identification process, which leads to more confident predictions overall. The photopeaks of $^{60}$Co are also at a higher energy than the last example; in general, fewer isotopes have peaks in this region, which reduces the probability that these peaks could have been caused by a different isotope.
Table 6.4: Isotope Posteriors for the Co-60 Spectrum

<table>
<thead>
<tr>
<th>Isotope</th>
<th>Presence Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-60</td>
<td>0.90034</td>
</tr>
<tr>
<td>Co-57</td>
<td>0.081167</td>
</tr>
<tr>
<td>I-125</td>
<td>0.042329</td>
</tr>
<tr>
<td>K-40</td>
<td>0.034162</td>
</tr>
<tr>
<td>Ba-133</td>
<td>0.033263</td>
</tr>
</tbody>
</table>

Figure 6.3: A background-subtracted 60 second spectrum of $^{133}$Ba and $^{152}$Eu, with apparent activities at the detector face of 0.2 µCi.

6.1.3 Demo: Ba-133 and Eu-152

While the previous simple cases are important, it is desired for the identifier to be able to identify more complicated sources. A background-subtracted spectrum containing sources $^{133}$Ba and $^{152}$Eu of approximately equal flux on the detector surface was measured with a 2 in x 2 in NaI detector, shown in Figure 6.3. The Bayesian classifier method was used to make the identifications.

As demonstrated in Table 6.5, the spectrum most likely contained a mixture of $^{133}$Ba and $^{152}$Eu. Table 6.6 shows the posterior probabilities for individual isotopes, which makes it clear that $^{152}$Eu is present. The probability that $^{133}$Ba is present is low (22.3%), though significantly elevated. Even on
Table 6.5: Model posteriors for the $^{133}$Ba and $^{152}$Eu spectrum.

<table>
<thead>
<tr>
<th>Isotope 1</th>
<th>Isotope 2</th>
<th>Isotope 3</th>
<th>Model Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ba-133</td>
<td>Eu-152</td>
<td></td>
<td>0.134</td>
</tr>
<tr>
<td>Ba-133</td>
<td>Th-232</td>
<td></td>
<td>0.079</td>
</tr>
<tr>
<td>Eu-152</td>
<td></td>
<td></td>
<td>0.039</td>
</tr>
<tr>
<td>Co-57</td>
<td>Eu-152</td>
<td>Th-232</td>
<td>0.035</td>
</tr>
</tbody>
</table>

Table 6.6: Isotope posteriors for the $^{133}$Ba and $^{152}$Eu spectrum.

<table>
<thead>
<tr>
<th>Isotope</th>
<th>Isotope Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eu-152</td>
<td>0.999</td>
</tr>
<tr>
<td>Th-232</td>
<td>0.424</td>
</tr>
<tr>
<td>Ba-133</td>
<td>0.223</td>
</tr>
<tr>
<td>Sm-153</td>
<td>0.085</td>
</tr>
</tbody>
</table>

its own, $^{133}$Ba has significantly overlapping peaks, and $^{152}$Eu has even more peaks in that region, making it harder to accurately detect those peaks.

In this example, the maximum model posterior was low (13.4%), but using the isotope posteriors very accurate identifications can still be made. The isotope posteriors clearly indicated that $^{152}$Eu is present but that other nuclides are not as confidently found.

6.1.4 Demo: Shielded Eu-152

The previous example used $^{133}$Ba, which is often a problem for low-resolution identifiers because of its significantly overlapping peaks at 276.4, 302.9, 356.0, and 383.8 keV, and $^{152}$Eu, a source with a very large number of peaks (156 peaks total), including many peaks that overlap with $^{133}$Ba (see Figure 4.7). However, the previous spectrum had many easily-resolvable peaks and no shielding (making accurate feature extraction much easier) and was background-subtracted (eliminates some of the continua and peaks from NORM isotopes, improving SNR). The next example again uses $^{152}$Eu, but introduces a 10 mm layer of lead between the source and the detector. As shown in Figure 6.4, most of the $^{152}$Eu peaks are eliminated by the lead. However, the classifier gives a very confident presence posterior for the correct source. Interestingly, adding shielding increased the europium posterior.
Figure 6.4: Spectrum of $^{152}$Eu with 10 mm of lead between the detector and source.

Eliminating lower energy peaks drastically reduced the number of candidate isotopes that could be present in the spectrum, which decreases the posterior probabilities for most other isotopes (as a vast majority of all of the gamma peaks in the library are in the $< 500$ keV region).

Table 6.7: Isotope posteriors for the shielded $^{152}$Eu spectrum.

<table>
<thead>
<tr>
<th>Isotope</th>
<th>Isotope Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eu-152</td>
<td>&gt;0.999</td>
</tr>
<tr>
<td>Xe-133</td>
<td>0.627</td>
</tr>
<tr>
<td>Ba-133</td>
<td>0.276</td>
</tr>
<tr>
<td>Tl-201</td>
<td>0.146</td>
</tr>
<tr>
<td>I-125</td>
<td>0.113</td>
</tr>
</tbody>
</table>

6.1.5 Demo: Weapons-Grade Plutonium

Arguably the most important requirement for handheld identifiers is that they must be able to identify special nuclear materials. First, a demonstra-
Figure 6.5: A 60 second spectrum of an unshielded weapons-grade plutonium source. A clear peak from $^{241}\text{Am}$ is also detected and properly identified.

Table 6.8: Model Posteriors for the WGPu Spectrum

<table>
<thead>
<tr>
<th>Isotope 1</th>
<th>Isotope 2</th>
<th>Isotope 3</th>
<th>Model Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Am-241</td>
<td>Pu-239</td>
<td></td>
<td>0.63403</td>
</tr>
<tr>
<td>Am-241</td>
<td></td>
<td></td>
<td>0.097537</td>
</tr>
<tr>
<td>Am-241</td>
<td>Ir-192</td>
<td></td>
<td>0.065025</td>
</tr>
<tr>
<td>Am-241</td>
<td>Tl-201</td>
<td></td>
<td>0.065025</td>
</tr>
<tr>
<td>Ir-192</td>
<td></td>
<td></td>
<td>0.059641</td>
</tr>
</tbody>
</table>

In this case, a correct identification of $^{241}\text{Am}$ and $^{239}\text{Pu}$ is produced (see Tables 6.8 and wgpuIP. While there are other isotopes present in this source (e.g. $^{237}\text{Np}$ and $^{240}\text{Pu}$), this identification is more than adequate for NaI detectors.

6.1.6 Demo: Shielded Highly-Enriched Uranium

In all likelihood, these field detectors are not going to encounter bare SNM lying in a field. Instead, it is more likely that these sources would be hidden in a car, or a shipping crate, etc. It is then strongly desired that these
Table 6.9: Isotope Postiors for the WGPu Spectrum

<table>
<thead>
<tr>
<th>Isotope</th>
<th>Isotope Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Am-241</td>
<td>0.88401</td>
</tr>
<tr>
<td>Pu-239</td>
<td>0.63944</td>
</tr>
<tr>
<td>Ir-192</td>
<td>0.12597</td>
</tr>
<tr>
<td>Tl-201</td>
<td>0.06956</td>
</tr>
<tr>
<td>U-238</td>
<td>0.03958</td>
</tr>
</tbody>
</table>

Figure 6.6: A 60 second spectrum of the Rocky Flats Shells, measured from 68 cm: 13.1 kg of highly enriched uranium (93.2% $^{235}\text{U}$, 5.4% $^{238}\text{U}$), with 0.5 inches of iron shielding surrounding the HEU source.

identifiers be able to identify SNM in the presence of significant shielding.

For the last demonstration, a 60 second measurement of a highly enriched uranium source wrapped in a half inch of iron is tested. This measurement was taken at the Nevada National Security Site in the Device Assembly Facility in 2015.

Despite the relatively few peaks visible in the spectrum, a correct identification of $^{235}\text{U}$ was made, as shown in Tables 6.10 and 6.11.

6.2 Preparing the Benchmarks

The DHS Algorithm Improvement Program (AIP) was created to give identifier manufacturers and research groups a way to evaluate their algorithms.
Table 6.10: Model Posteriors for the Shielded HEU

<table>
<thead>
<tr>
<th>Isotope 1</th>
<th>Isotope 2</th>
<th>Isotope 3</th>
<th>Model Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-235</td>
<td></td>
<td></td>
<td>0.68505</td>
</tr>
<tr>
<td>Ra-226</td>
<td></td>
<td></td>
<td>0.27502</td>
</tr>
<tr>
<td>Am-241</td>
<td>U-235</td>
<td></td>
<td>0.022961</td>
</tr>
<tr>
<td>U-238</td>
<td></td>
<td></td>
<td>0.001459</td>
</tr>
<tr>
<td>Ir-192</td>
<td></td>
<td></td>
<td>0.00123</td>
</tr>
</tbody>
</table>

Table 6.11: Isotope Posteriors for the Shielded HEU

<table>
<thead>
<tr>
<th>Isotope</th>
<th>Presence Posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-235</td>
<td>0.71174</td>
</tr>
<tr>
<td>Ra-226</td>
<td>0.27711</td>
</tr>
<tr>
<td>Am-241</td>
<td>0.023025</td>
</tr>
<tr>
<td>Ba-133</td>
<td>0.008607</td>
</tr>
<tr>
<td>I-125</td>
<td>0.005073</td>
</tr>
</tbody>
</table>

and to compare them against each other. This program is in the beta testing stage. The benchmarking process has three parts: detector modeling, spectral simulation, and quantifying performance.

While it would be preferable to use exclusively measured (non-simulated) spectra for algorithm evaluation, it is not currently feasible for this research group due to limited access to radiation sources, particularly SNM and medical isotopes. The spectral simulation code described in Chapter 4 could be used for this, but the DHS AIP produces much more realistic spectra, as shown in the Section 6.2.2. The two spectral simulation codes actually use approximately the same approach; however, the implementation described in this dissertation used some experimentally measured detector characterization, while GADRAS-DRF computes them itself by more elaborate detector modeling described in the following section. The GADRAS-DRF fits will actually be significantly better, as there are many parameters that were not reasonably experimentally observable, such as the characteristics of the attenuating materials within the detector.

6.2.1 Detector Modeling

To begin, a detector model must be created. This uses GADRAS-DRF (GAmma Detector Response and Analysis Software- Detector REsponse Func-
tion), a public version of GADRAS. GADRAS is a large suite of software that has been developed by Sandia National Laboratories. GADRAS itself has a wide range of capabilities, from detector modeling to radiation transport [88, 87]. The publicly-available GADRAS-DRF is a subset of GADRAS that has advanced capabilities like radiation transport, inverse transport calculations, and other analyses removed [87].

Within GADRAS-DRF, a set of calibration spectra (containing $^{60}\text{Co}$, $^{133}\text{Ba}$, and $^{137}\text{Cs}$, plus background measurements) is used to build the detector model. A menu of the detector parameters is shown in Figure 6.7, and the associated fits are shown in Figure 6.8. While the fits are not perfect, they are significantly better than the simulation code developed as a part of this dissertation, particularly at lower energies.

![Figure 6.7: The detector menu in GADRAS-DRF, which shows all of the parameters used for the detector model. While user estimates can be used, GADRAS-DRF has parameter optimization built in so that better models can be built.](image-url)
Figure 6.8: Comparison of the measured calibration spectra overlaid with the GADRAS-DRF fit. The fits here underestimate the Compton scattering and overestimate the peak height; this is likely due to the nearby presence of dense scattering material in the calibration spectra, such as the ground or a metal table. For this reason, the documentation recommends suspending the detector and source in mid-air if feasible [89].

6.2.2 Spectral Simulation

GADRAS-DRF is combined with a set of Excel files and Python code to produce simulated spectra. Inject files are created with the Excel/Python files, and GADRAS-DRF performs the actual spectral computations.

For the simulation, sources were chosen from a provided list. The sources here have been carefully measured in several configurations and deconvolved with the detector response function. They can then be convolved with the desired detector response function to produce appropriate templates. This method can generate accurate spectra, but it has the disadvantage of only allowing for the simulation of a limited set of detection scenarios. The AIP
also does not currently allow for the simulation of mixed radiation sources (aside from background).

Terrestrial and cosmic radiation modeling is included in this simulation. The user can choose a background from many locations (e.g. Chicago Airport, though no distinction is made between O’Hare and Midway) or specify the desired isotopics for background simulation.

For the benchmarking here, a set of sources was chosen to try to use the ANSI N42.34 standard, as described previously. All of the sources used were unshielded, and were simulated at full source strength for a 60 second measurement time. The specific sources used are detailed in Table 6.12. The primary difference between this benchmark and the ANSI N42.34 standard is that this simulation uses HEU instead of separate $^{235}$U and $^{238}$U sources and WGPu instead of reactor-grade plutonium, due to the currently available sources in the beta-version of this simulation suite.

Table 6.12: AIP sources used for benchmarking.

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Source ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Am-241_Unshielded</td>
<td>1001_AIP</td>
</tr>
<tr>
<td>2</td>
<td>Ba-133_Unshielded</td>
<td>1121_AIP</td>
</tr>
<tr>
<td>3</td>
<td>Co-57_Unshielded</td>
<td>2002_AIP</td>
</tr>
<tr>
<td>4</td>
<td>Co-60_Unshielded</td>
<td>1021_AIP</td>
</tr>
<tr>
<td>5</td>
<td>Cs-137_PE-11.589-cm</td>
<td>1043_AIP</td>
</tr>
<tr>
<td>6</td>
<td>Eu-152_Unshielded</td>
<td>1217_AIP</td>
</tr>
<tr>
<td>7</td>
<td>Ga-67_Unshielded</td>
<td>2025_AIP</td>
</tr>
<tr>
<td>8</td>
<td>HEU_Unshielded</td>
<td>4001_AIP</td>
</tr>
<tr>
<td>9</td>
<td>I-125_Photom_LowAtten</td>
<td>2345_AIP</td>
</tr>
<tr>
<td>10</td>
<td>I-131_Unshielded</td>
<td>2049_AIP</td>
</tr>
<tr>
<td>11</td>
<td>Ir-192_Unshielded</td>
<td>4013_AIP</td>
</tr>
<tr>
<td>12</td>
<td>K-40_NoSalt_Unshielded</td>
<td>1055_AIP</td>
</tr>
<tr>
<td>13</td>
<td>Ra-226_Unshielded</td>
<td>3015_AIP</td>
</tr>
<tr>
<td>14</td>
<td>Tc-99m_Unshielded</td>
<td>3119_AIP</td>
</tr>
<tr>
<td>15</td>
<td>Ti-201_Unshielded</td>
<td>2096_AIP</td>
</tr>
<tr>
<td>16</td>
<td>U-233_Unshielded</td>
<td>2104_AIP</td>
</tr>
<tr>
<td>17</td>
<td>WGPu_Unshielded</td>
<td>4011_AIP</td>
</tr>
</tbody>
</table>
6.2.3 Performance Metrics

The DHS AIP uses a weighted F-Score (also called an $F_1$ score) to measure an algorithm’s performance [90]. This accounts for both the precision and recall of an algorithm, as well as the political importance of a particular isotope.

The precision $p$ is the probability that a positive identification is correct. If $t_p$ is the true positive rate and $f_p$ is the false positive rate, the precision $p$ is

$$p = \frac{t_p}{t_p + f_p} \quad (6.1)$$

Similarly the recall $r$ is the probability that an isotope will be identified if it is present. If $t_p$ is the true positive rate and $f_n$ is the false negative rate, the recall $r$ is then:

$$r = \frac{t_p}{t_p + f_n} \quad (6.2)$$

Finally, the F-score is the harmonic average of the precision and recall:

$$F = 2 \cdot \frac{p \cdot r}{p + r} \quad (6.3)$$

Table 6.13 shows some example F-score calculations. F-scores are strongly biased towards the lower value; this is useful for isotope identification applications, as an algorithm that always identifies every isotope is present will have perfect recall (1.0) but minimal precision, leading to an overall low score.

The DHS AIP uses both the unweighted F-score and weighted F-scores. For obvious nuclear security reasons, some nuclides are much more important to identify accurately than others, such as weapons-grade plutonium.
(WGPu). To account for this, each nuclide in the scoring application is given a weight factor (WF) of 3, 2, or 1 for high, medium, or low importances respectively (although this is modifiable by the user so that the user can optimize for specific applications). For the computation of a weighted F-score, a high importance isotope would count as three correct or missed identifications, etc.

It is important to note that low-resolution detectors and the simulation parameters presented in this dissertation (most notably, count-time and background levels) are expected to cause a significant reduction in score. The answer key used in the nuclide scoring application was made using HPGe detectors with long count times, so there are isotopes in the answer key that will not be detectable with an NaI detector with these short count times [91]. For this reason and others, the maximum possible score for low-resolution detectors is not known. However, this is still useful for optimization and for comparing the performance of a set of algorithms on a given training set.

6.3 Algorithm Benchmark Performance

For this benchmarking process, two similar sets of spectra were created with the AIP software. The first was used to optimize the performance, and the second was used for performance evaluation. However, due to the count time and measurement configurations, the performance between these two data sets are extremely similar. Improvements to this process are discussed in Chapter 7. Example spectra of each of the sources used are shown in Figure 6.9.

For the optimization process, the various parameters in the feature extraction and identification code were individually modified to find the best performance on the training benchmark set. In general, there are better methods for optimizing over several parameters that would yield better final identification performance. However, the current scoring routine in the AIP can only evaluate one results set at a time and only through the graphical user interface. Future versions of this scoring application should have the ability to score multiple results sets at once, which would greatly improve the optimization process.

The identification performance on the validation set is shown in Figure
Figure 6.9: Example spectra for each of the sources used in the benchmark. All of these were simulated in GADRAS-DRF using the default source strengths at a distance of 25 cm from the 2”x2” NaI detector.
6.10. An unweighted F-score of 66.8 and weighted F-score of 66.33 were achieved. The associated precision and recall rates are approximately 67% and 69% respectively.

![Mean F-Score, Precision, & Recall Bar Charts](image)

Figure 6.10: Algorithm performance on the AIP unshielded test set, without using background subtraction.

6.4 Performance Discussion

While precisions and recalls on the order of 65 – 70% are an improvement over published detector evaluations, it is desired to improve these further. Methods for future algorithm extensions and modifications are discussed in Chapter 7, but some issues with this benchmarking are discussed here.

First, the activities of the sources in this benchmark are not known. From the spectra in Figure 6.9 shown previously, the count rates of these sources are on the same order of magnitude. However, the ANSI standard has specific requirements for exposure rates. This could be accounted for when generating the simulated spectra for the benchmark but not until the source information is provided.

Second, there are specific sources in this benchmark that challenge the identification algorithm. These challenges are primarily due to the feature extraction code used, in the answer key issue as discussed in Section 6.2.3, or in the model library.
• $^{125}$I decays via electron capture and emits only very low energy gamma-rays, with the highest at 35.5 keV [35]. The peak at 35.5 keV is just above the lower level discriminator (a low energy cutoff), and the feature extraction algorithm doesn’t reliably recognize it as a photopeak; it is correctly found in just two of the ten trials. Thus, for most tests no peak information suggesting that $^{125}$I is present is passed to the identification code, and it is never identified. Solving this issue with the feature extraction code would raise the F-scores by approximately 5%.

• $^{152}$Eu is accurately identified by this algorithm, but the source also contains trace amounts of $^{154}$Eu. Not only are these peaks not detected due to their very low area, $^{154}$Eu is not in our library, making its identification impossible. This results in a partially correct score for this source. This issue occurs for a few of the sources in this benchmark (e.g. only $^{241}$Am and $^{239}$Pu are identified in the WGPu source, while $^{232}$U and $^{241}$Pu are expected by the AIP scoring key).

• $^{226}$Ra was often misidentified or identified along with an incorrect isotope in this benchmark. This is again primarily a library issue specific to this benchmark. The model library used only contains a pure $^{226}$Ra source. However, these sources generally have significant contributions from its daughters. This can be remedied by including an “effective” model for the radium decay chain, where the contributions from the daughter isotopes are included. This has been done for other model libraries, but the extended model was not used for this benchmark.

• $^{233}$U is not identified correctly in any of the tests. This is due to a shortcoming in the model library. $^{233}$U is bred by neutron irradiation of $^{232}$Th. This process results in a contamination of $^{232}$U, which leads to emissions from several other sources ($^{228}$Th, $^{224}$Ra, $^{220}$Rn, $^{216}$Po, $^{212}$Pb, $^{212}$Bi, and $^{208}$Tl). These contributions could be accounted for in future version of the model library, which should result in correct identifications.

If the three examples of problems with the model library above were fixed and correct identifications were obtained for these two sources, the F-scores would be improved to approximately 78.8%. If both the library issues and the
feature extraction at low energy were fixed, F-scores of approximately 84.5 would be obtained. This corresponds to unweighted precisions and recalls of 88.7% and 82.2% respectively. In each case, fixing one of the mentioned problems results in an improvement of 4 – 7%.

Once these two problems are solved, this algorithm would be correctly identifying every source in this benchmark in most trials. With some final optimization, it is feasible for this approach to meet the ANSI standard for handheld RIIDs, even if it does not currently.

6.5 Limits of Detection

To understand the limits of detectability, the DHS AIP software was used to simulate ten $^{60}$Co spectra for each measurement length (1 s, 3 s, 5 s, 10 s, 20 s, and 40 s) with the default source strength and the “Chicago, IL” radiation background (1.34% $^{40}$K, 1.99 ppm U, 5.24 ppm Th, 180 m above sea level). The total background count rate as measured by the detector was 121 cps, while the source is 772 cps. Example spectra used for identification are shown in Figure 6.11, as well as the corresponding background spectra for reference.

For this evaluation, “Correct” means that $^{60}$Co was identified either alone or together with a background isotope (e.g. $^{40}$K). “Partial correct” means that $^{60}$Co was identified along with an incorrect isotope (e.g. $^{99m}$Tc), and “wrong” means that $^{60}$Co was not identified. In addition, the precision, recall, and F-score were calculated using the DHS AIP metrics.

Performance of this algorithm is presented in Table 6.14. Because this algorithm uses peak-based identification, it is unsurprising that it has poor performance with short measurement times, such as the 50% of one second trials it failed to identify $^{60}$Co. The identification performance improves with the counting statistics because the feature extraction becomes more accurate.

However, the detection limits are highly situational. First, the actual radiation source matters tremendously, even with similar counting statistics. Certain photopeaks are easier to reliably detect than others; for example, the 1173 and 1332 keV peaks of $^{60}$Co rarely overlap with other photopeaks and are at a high enough energy to usually have little background underneath them. Some isotopes are also easier to identify than others, such as $^{152}$Eu.
Figure 6.11: $^{60}$Co and background spectra simulated for the limits of detection experiment. Ten spectra of each measurement time were used for this analysis. Background subtraction was not used for this test.
Table 6.14: Detection Limits of Co-60 Spectra

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Correct</th>
<th>Partial correct</th>
<th>Wrong</th>
<th>Precision</th>
<th>Recall</th>
<th>Fscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30%</td>
<td>20%</td>
<td>50%</td>
<td>40</td>
<td>50</td>
<td>43.33</td>
</tr>
<tr>
<td>3</td>
<td>40%</td>
<td>20%</td>
<td>40%</td>
<td>48.33</td>
<td>60</td>
<td>51.67</td>
</tr>
<tr>
<td>5</td>
<td>40%</td>
<td>40%</td>
<td>20%</td>
<td>66.67</td>
<td>66.67</td>
<td>60</td>
</tr>
<tr>
<td>10</td>
<td>40%</td>
<td>40%</td>
<td>20%</td>
<td>63.33</td>
<td>73.33</td>
<td>63.33</td>
</tr>
<tr>
<td>20</td>
<td>40%</td>
<td>40%</td>
<td>20%</td>
<td>58.33</td>
<td>80</td>
<td>65</td>
</tr>
<tr>
<td>40</td>
<td>70%</td>
<td>20%</td>
<td>10%</td>
<td>83.33</td>
<td>83.33</td>
<td>80</td>
</tr>
</tbody>
</table>

(which has many peaks across a wide energy range) or $^{60}$Co (which has few peaks but is not easily mistaken for other isotopes). Second, the radiation background will play a large role in the detection limits and will affect the identification of some sources more than others. A higher radiation background will make the photopeaks of interest harder to detect, particularly if the photopeaks of interest are near the background peaks or Compton edges.

To rigorously understand the detection limits of the algorithm, the usage scenarios should be well-defined, and a study similar to this one should be completed with a wide range of sources and shielding configurations.
7.1 Conclusions and Future Work

A different approach to automated isotope identification has been developed and demonstrated in this dissertation. This approach uses peak energy and area information to make identifications, but is modular in its approach and could include other information as well. While it is conceptually similar to library comparison methods, this approach is capable of dealing with poorly calibrated data, with complicated mixtures of isotopes, and with unknown shielding materials. Tests of this algorithm continue to show that it is a viable approach for identification. Before commercial deployment could be considered, there are many areas to improve this algorithm, as well as related methods to explore.

7.2 Improving the Feature Extraction

To continue development of this methodology, the first step is to implement a better feature extraction method. It may be possible to improve the current code enough to reach desired performance, but this goal will likely require the exploration of other feature extraction methods.

Currently, the wavelet/NNLS code is the primary limiting factor in the identification performance. This can be readily verified by comparing identification performance on manually-identified peaks against the wavelet/NNLS results. No feature extraction method is going to be flawless for low-resolution handheld detectors, but an improved peak detection capability is critical for this project to advance. As discussed in the last chapter, the wavelet code/NNLS also have particular difficulties in detecting low-energy peaks,
which needs to be remedied as well.

The wavelet/NNLS code is also the limiting factor in the computation time, running in approximately 30 seconds versus 2 seconds for the identification code (both in MATLAB on the same machine). If a wavelet-based methodology is to be used in future versions, a hardware-based wavelet method could be implemented to significantly reduce the computation time.

It is possible that the feature extraction could be improved while still using the current method. In particular, one assumption in this method is that the wavelet basis set can be found using perfect Gaussians. However, as demonstrated in Section 4.4, the detector response is certainly not Gaussian overall, which breaks this assumption. This is especially a problem when trying to resolve overlapping peaks, as the Compton edges can interfere with this reconstruction. Better basis functions could be computed on the fly, but this would have the unfortunate side-effect of increasing computation time further.

Alternatively, the wavelet/NNLS could be replaced entirely. There are many other peak detection and quantification methods, such as a neural network approach. Some of these methods could potentially be used to detect other spectral features, such as Compton edges and bremsstrahlung continua cutoff energies; this could be used to improve the detector calibrations, or even to aid in the identification process.

7.3 Improving the Model Library

As noted in Chapter 6, shortcomings of the model library are the other most significant limitation on this method’s performance.

First, a model in the library is either one, two, or three isotopes together. While this is suitable for most problems, the library needs to contain some decay chain and common contaminant sources. In particular, the inclusion of a more realistic $^{233}\text{U}$ source would make a significant impact on performance.

Second, a study should also be done to determine precisely how large the model library should be for accurate identification. As described in Chapter 4, the library contains combinations of isotopes in certain activity ratios. It may be the case that the number of activity ratios could be reduced without impacting performance (which would reduce the computation time required),
or that a different set of activity ratios would improve performance.

7.4 Improving the Bayesian Classifier

There are many methods to improve the peak-based Bayesian classifier presented here. While all of the scoring functions could likely be improved, the prior probabilities and peak pairing have significant room for improvement. Further, incorporating other data sources could make a significant difference in performance.

7.4.1 Prior Probabilities

For this dissertation, only two priors were studied. There are many other priors that would be interesting to explore.

First, one viable prior would be to have a manually-defined prior for each combination of isotopes. This could eliminate some current bad test cases, where an isotope like $^{204}\text{Tl}$, a medical isotope, is identified with weapons-grade plutonium (WGPu). In reality, WGPu spectra usually contains $^{241}\text{Am}$, which produces a large peak at 59.5 keV; $^{204}\text{Tl}$ has a large peak at 70.8 keV, and if the detector is even slightly miscalibrated the observed peak could reasonably be attributed to $^{204}\text{Tl}$. A human-defined prior could eliminate problems such as these by setting a very low prior for nonsensical mixtures like this, but this would be very time-consuming even for small isotope libraries.

Another possibility would be to ship these handheld detectors with a non-informative prior and have them learn a better prior as they are used. This would have the advantage of creating custom priors for each detector deployment. However, extreme caution would be necessary here. First, the identifications from these RIIDs are not always right, so using the identifications to learn a prior is somewhat problematic. Second, if a particular source (e.g. WGPu) was almost never encountered, a learned prior for it would be very low, which would make it harder to detect.
7.4.2 Peak Pairing

Peak pairing was done with a modified nearest neighbor lookup, with a slight dynamic recalibration. However, there are other methods that could be used, and the current methods would likely not work for much higher resolution detectors.

One could also try choosing the calibration that maximizes the whole likelihood function \( P(D|M) \) instead of just the peak position score. This could yield significantly better matches, as this would also consider the peak areas. However, with the models used in this dissertation, this maximization is not nearly as straightforward as maximizing just the peak position score because the other scores are not differentiable, and this optimization would result in a greatly increased computation time.

It is unlikely that this would make much of a difference on low-resolution detectors. Because peaks that are “close” together cannot be resolved, bad calibrations don’t always change the peak pairs. For these low-resolution detectors, this would likely only impact very poorly calibrated detectors with multiple peaks in the low energy region.

However, for higher resolution detectors, mistakes in the detector calibration would be more significant due to the increased observable peak density. Compared to the NaI detectors used here, higher-resolution detectors will (by definition) be able to resolve many more peaks, resulting in a greatly increased peak density. A poor detector calibration will then cause many peaks to be mismatched in this step, resulting in degraded performance. NaI detectors, combined with the wavelet/NNLS peak detection algorithm, do not yield many peaks close in energy, and this mismatching is less of a concern.

7.4.3 New Data Sources

Other likelihood model possibilities were discussed in Section 5.4.8. In particular, the inclusion of a neutron detector could be used to quickly improve performance. If neutrons are detected at all, a significant penalty could be applied to any model in the library that does not produce neutrons. This would easily improve the performance without increasing the computational complexity of the methodology.

Besides incorporating new hardware sources of data, other features could
be obtained from the spectrum. Beta decay information, Compton edges, and escape peaks could all be used in the identification process.

7.5 Improving the Spectra Simulation

Moving forward, there are two primary areas to improve this method. First, detector response functions for more model geometries and shielding configurations should be generated. This will be critical for the neural network projects, as they must be able to cope with shielding materials. Second, some of the detector characteristics used for the MCNP simulations should be modified. In particular, the MCNP simulations used energy bins with constant linear steps. However, the energy bins of these detectors grow non-linearly with energy, so these bins need to be updated. This is a simple change (one to two lines of code), but was not necessary for the applications within this dissertation.

However, tools have been written to automate most of this process, and a complete set of response functions for one source/shielding/detector geometry can be generated in approximately 16 hours on a desktop with an i7-5820k processor. The MCNP simulation time could be significantly reduced with some of the internal variance reduction methods, but this was not necessary for this implementation and has not been used. New response functions with shielding material will be computed in the near future (see Section 4.4.4).

7.6 Improving the Benchmarking and Optimizations

There are several improvements to be made with the AIP application for optimization and benchmarking. Some of these are changes for the program administrators to implement, while others are on the user side.

On the administrator side, more source information needs to be provided to the user. This is critical for future performance evaluations, as detecting a 1 \(\mu\)Ci source instead of a 1 Ci source will be much more difficult, etc. It would also help tremendously to be able to batch the evaluations instead of manually pasting in one set at a time; this would allow for much greater optimization. It will also be necessary in the future to allow for the generation
of mixed radiation sources, as this is a requirement for these detectors and is not currently implemented within the AIP.

On the user side, there are a few ways to improve the detector modeling, although they are not important for this application. First, more calibration spectra could be used. In particular, a low-energy source like $^{241}\text{Am}$ and a high energy source like $^{228}\text{Th}$ would provide important calibration points at the extreme energy ranges, leading to better detector characterization. Second, the calibration spectra could be collected under more agreeable conditions. Placing the sources and the detector further from dense scattering materials would make the peak to Compton ratio fits agree more closely with the experimental observations.

Beyond improving the modeling, improving the optimization will require using a wider range of isotopes and using mixtures of isotopes. Future work will also need to examine the sensitivity to the parameters as measurement time is adjusted (e.g. does the same parameter set work for 15 second spectra versus 60 second spectra?).

7.7 Other Applications

Besides trying to improve performance of this algorithm on the same NaI detectors, there are other applications of this methodology that should be explored.

First, this algorithm should be tested with other types of detectors, particularly medium-resolution detectors. This would require only some minor modifications to the likelihood modeling and another round of parameter optimization. This could potentially greatly improve the identification performance.

Second, there are similar methods to the peak-based Bayesian classifier that could be studied. Instead of constructing a peak-based classifier, a region of interest method could be developed instead. The Bayesian classifier scheme could eliminate the need for non-overlapping ROIs and could better incorporate ROI counts. Because there would be no need for feature extraction, the total computational requirements would be greatly reduced. This opens up the possibility of doing real-time or near real-time identification updates. Unlike the peak-based approach in this dissertation, this ROI
approach could be developed for very low-count spectra.

A different approach would be to find a new representation of the spectrum. This could be done in many different ways, such as one of the deconvolution methods discussed in Chapter 3 or with an autoencoder. These techniques have been applied to other spectroscopic problems, such as material identification with hyperspectral imaging [92] or optical emission spectroscopy [93], but not gamma-ray spectroscopy for radionuclide identification. Identification could then be performed with this new spectral representation. This would require building possibly entirely different likelihood scoring functions, but this is another option for performing identifications.
REFERENCES


Presented here is a cut down version of the identification code. This code assumes the following directory structure.

Isotope Identification Project/
  ____BRAIN/
    ___computePosteriors.m
    ____f_AR.m
    ____f_DPI.m
    ____f_LPI.m
    ____f_PP.m
    ____logicLikelihood.m
    ____makeOutput.m
  ____Library/
    ___LibraryGeneration/
      ___Isotopes/
        ____Detector/
          ___DetectorEfficiency.m
          ___DetectorFWHM.m
          ___simple_GenerateLibrary.m
        ___makeIsotopeIndex.m
        ___makeNameIndex.m
        ___reducemodel.m
        ___ModelLibrary_ANSIonly_Gauss.mat
  ____Misc Functions and Scripts/
    ___readCHN.m
  ____Tests/
    ___Wavelets/
    ___setup_BRAIN.m
    ___run_identifications.m

These MATLAB functions and scripts are presented here, excluding the Wavelet directory and the readCHN.m function. The Tests/ directory contains various spectra for demonstrating the code.
A.1 Isotope Identification Project/

A.1.1 setup_BRAIN.m

This script will create all directory paths and loaded the data files needed for identification.

```matlab
% Load everything needed for identification with BRAIN
%
% This script will add all of the filepaths needed and will...
% load all of the
% data sets needed to use the Bayesian RADionuclide ...
% Identification code.
if exist('BRAINLoaded')==0
    % Add necessary directories to the path
    % ID Code
    addpath(genpath('BRAIN'));
    % Library data
    addpath(genpath('Library'));
    % Feature extraction code
    addpath(genpath('Wavelets'));
    % Spectra for testing
    addpath(genpath('Tests'));
    % readCHN, etc
    addpath(genpath('Misc Functions and Scripts'));
    addpath(genpath('Dissertation_Benchmarking'));
    addpath(genpath('Dissertation_Plots'));
    %Load the model library
    load('Library/ModelLibrary_ANSIonly_gauss.mat');
    makeNameIndex;
    makeIsotopeIndex;
    BRAINLoaded=1;
end
format shortg;
```
This script demonstrates the identification process from start (a calibrated spectrum) to finish (model and isotope posteriors).

```
% BRAIN Isotope identification code
%% run_identifications.m
% This script will perform identifications from start (... spectrum) to finish
% (identifications). This script will:
% 0. Load everything needed for ID and define ...
% identification parameters.
% 1. Load the spectrum and calibration to process.
% 2. Run the feature extraction code on the spectrum.
% 3. Run the identification code.
% 4. Output (to the console) the ID results.

% 0. Load everything needed for ID and define ID ...
% parameters.
% Load everything, initialize directories
close all;
setup_BRAIN;

% define ID parameters. These are stored in a structure ...
% and passed
% throughout the code.
IDparams.wmin   = 15;
IDparams.wmax   = 60;
IDparams.thetaDPP = 0.1;
IDparams.thetaLPP = 0.05;
IDparams.UncertaintyThreshold=1.8;

% 1. Load the spectrum and calibration.
% spectrum:= a 1024x1 vector of count data. This can be ...
% read from a CHN
% file with readCHN.m, or can be defined manually in other ...
% ways (e.g. read
% from a csv or a mat file, etc).
% Example:
spectrum=readCHN('Tests/Clair_LANL/Cs137@25cm.Chn');
spectrum=spectrum.count;
```
32  \% spectrum=xlsread('BERP_60s_Unshielded.xlsx');
33  \%
34  \% The calibration is stored as a structure.
35  \% E= ch^2 * cal.a + ch * cal.b + cal.c
36  cal.a=0.0001;
37  cal.b=3.2779;
38  cal.c= -17.734;
39  \%
40  \% 2. Run the feature extraction code on the spectrum.
41  \% The output of whatever feature extraction code should be ...
42  \% an nx3 (or nx2)
43  \% matrix, where each row is [Energy_i Area_i Uncertainty_i]...
44  waveletresults=WaveletCodeLoud(spectrum,cal);
45  \%
46  \% The output of the wavelet code needs one additional ...
47  \% filter:
48  \% Throw out all peaks with uncertainty > ...
49  IDparams.UncertaintyThreshold
50  spectrumData=[];
51  for i=1:length(waveletresults.unc)
52      if waveletresults.unc(i) / waveletresults.area(i) < ...
53          IDparams.UncertaintyThreshold
54              spectrumData=[spectrumData; waveletresults.centroid... 
55                  (i), waveletresults.area(i),waveletresults.unc(... 
56                  i)/waveletresults.area(i)];
57          end
58  end
59  spectrumData
60  \%
61  \% 3. Run the identification code.
62  \% 3a. Define the Prior function.
63  \% uniform prior function
64  priorf=@(n,k) 1/(nˆk);
65  \%
66  \% 3b. Compute the likelihood score for each model in the ...
67  nmodel=size(ModelLibrary,2);
68  pv=zeros(nmodel,1); \% list of likelihood scores for each ...
69  model
70  for i=1:nmodel
71      if isempty(ModelLibrary(i).lib)==0
72          model=ModelLibrary(i).lib;
% Remove peaks below 35 keV, if they even exist.
% This is probably no longer needed with the updated library.
model=reducemodel(model);
if isempty(model)==0
    pv(i)=logicLikelihood(spectrumData,model,...
        IDparams)*priorf(30,size(ModelLibrary(i)... .name,1));
end
end

%% 4. Output (to the console) the ID results.
makeOutput
A.2 BRAIN/

These functions and scripts perform the computations for the Bayesian classifier and create the desired output.

A.2.1 computePosteriors.m

```matlab
%% This computes the combination posteriors for everything ... up to 3-source combinations.
% Eventually, the library could be combined into one ... structure and all of 
% this could be simplified to a dozen or less lines, but ... for the moment I 
% prefer to be able to split it up.

% ratstep=1; % change to 1 to test everything in the ... library. For higher values, skips some of the library ... for testing/speed purposes. Do not recommend.

% uniform prior function
priorf=@(n,k) 1/(n^k);

%%
nmodel=size(ModelLibrary,2);
pv=zeros(nmodel,1);
tic;
empties=0;
for i=1:nmodel
    if isempty(ModelLibrary(i).lib)==0
        model=ModelLibrary(i).lib;

        % Remove peaks below 35 keV, if they even exist. % This is probably no longer needed with the ... updated library.
        model=reducemodel(model);

        % This check should be obsolete with the updated ... library.
        if isempty(model)==0
```
27 pv(i)=logicLikelihood(spectrumData,model,...
   IDparams)*priorf(30,size(ModelLibrary(i)...
   .name,1));
28   end
29   else
30       empties=empties+1;
31   end
32 
33 end

A.2.2 f_AR.m

1 function [ outf ] = f_AR( matchedmat )
2 %F_AR Computes the area ratio probability
3 
4 nmatch=size(matchedmat,1);
5 librat=zeros(nmatch,1);%more efficient for create an nx1 ...
   matrix here and add values later
6 datrat=zeros(nmatch,1);
7 for i=2:nmatch
8     librat(i)=matchedmat(i-1,4)/matchedmat(i,4);
9     datrat(i)=matchedmat(i-1,2)/matchedmat(i,2);
10    end
11 
12 outf=1;
13 
14 for i=2:nmatch
15     librat=matchedmat(i-1,4)/matchedmat(i,4);
16     datrat=matchedmat(i-1,2)/matchedmat(i,2);
17     leftb=librat*exp(-matchedmat(i,1)/3000);
18     rightb=1.1*librat;
19     if datrat<rightb
20         if datrat<leftb
21             out=1+exp(-leftb)-exp(-datrat); %too left of ...
22                 ideal
23         else
24             out=1; %datrat is within tolerance of ideal
25         end
26     end
else
    out=exp(-1*(datrat-rightb));
end
outf=outf*out;
end

A.2.3 f_DPI.m

function [ out ] = f_LPI( lib,libflag )
%F_DPI Probability of this % of DATA peaks to be identified
tmp=lib(:,2);
total=sum(tmp);
identified=0;
for i=1:size(libflag,1)
    identified=identified+tmp(i)*libflag(i);
end
out=identified/total;
end

A.2.4 f_LPI.m

function [ out ] = f_DPI( data,datflag )
%F_DPI Probability of this % of DATA peaks to be identified
tmp=data(:,2);
total=sum(tmp);
identified=0;
for i=1:size(datflag,1)
    identified=identified+tmp(i)*datflag(i);
end
out=identified/total;
% function [ out ] = f_PP( matchedmat )
% %F_PP Computes the probability of the energy match
% % Uses more realistic prob function
% %THIS DOES NOT PENALIZE THE MISSED PEAKS AGAIN, UNLIKE ...
% THE ORIGINAL CODE
%
% nmatch=size(matchedmat,1);
% datE= matchedmat(:,1);
% libE=matchedmat(:,3);
%
% out=1; %initialize output probability
% for i=1:nmatch
%   pmin=.25-.2*(datE(i)/3000);
%   pdist=abs(datE(i)-libE(i));
%   wmin=10;
%   wmax=30;
%   w=wmin+(datE(i)/3000) *(wmax-wmin);
%   out=(pmin+(1-pmin) *cos((pi/2)*pdist/w)ˆ2)*out;
%   sigma=10;
%   out=normpdf(pdist,0,sigma);
%   out=out*exp(-pdist^2 / (2*sigma^2));
% end
%
% end
%
% function [ out ] = f_PP( matchedmat )
% %F_PP Computes the probability of the energy match
% % Uses the correlated peak score
% % Last modified 03/31/2016
nmatch=size(matchedmat,1);
datE=matchedmat(:,1);
libE=matchedmat(:,3);
sigma=10;

% Calculate the scale parameter s
numerator=0;
denominator=0;
for i=1:nmatch
    numerator=numerator+datE(i)*libE(i)/(sigma^2);
    denominator=denominator+datE(i)*datE(i)/(sigma^2);
end
s=numerator/denominator;

% out=1; %initialize output probability
% Calculate it more efficiently
out=exp(-(s*datE-libE)' * (s*datE-libE)/(2*sigma*sigma));

% for i=1:nmatch
%   % pmin=.25-.2*(datE(i)/3000);
%   % pdist=abs(s*datE(i)-libE(i));
%   % wmin=10;
%   % wmax=30;
%   % w=wmin+(datE(i)/3000)*(wmax-wmin);
%   %
%   % out=(pmin+(1-pmin)*cos((pi/2)*pdist/w)ˆ2)*out;
%   %
%   % out=normpdf(pdist,0,sigma);
%   % out=out*exp(-pdist^2 / (2*sigma^2));
% end
end
function [ out,libflag,dataflag ] = logicLikelihood( data, ... 
libisotope,IDparams)

% data should be in two columns
% Updated models based on validation experiment
% Rewritten method using modular functions.
% Performs peak matching and calls functions to calculate ...
% likelihood
% load('meshstruc.mat');

%% parameters
% wmin=15; % min neighborhood , was 5
% wmax=60; % max neighborhood, was 25
% thetaPP=0.05;% missing peak penalty

wmin=IDparams.wmin;
wmax=IDparams.wmax;
thetaDPP=IDparams.thetaDPP;
thetaLPP=IDparams.thetaLPP;

%% Useful numbers
lib=libisotope; %need a copy, going to delete used elements...
% later
nlib=size(lib,1); %number of peaks predicted by libisotope
ndata=size(data,1); %number of peaks detected

%% peak matching
1. Perform the matching. Create a matrix [E_D A_D E_L A_L...
], a library
%vector of flags to indicate whether an element was used, ...
% and a similar
%data vector. 1=used, 0=not. Not necessary, but useful ...
% for quickly
%testing new code. Will slow down the code somewhat.

[ matchedmat,dataflag,libflag ]=peakmatching(lib,nlib,data,...
data, wmin,wmax);
%% Compute the likelihood
ndatamissed=ndata-sum(dataflag); % number of missed data peaks
nlibmissed=nlib-sum(libflag); % number of missed library peaks
PPpenalty=thetaPPˆ(ndatamissed);
dataPPpenalty=(thetaDPP)ˆndatamissed; % .005
libPPpenalty=(thetaLPP)ˆnlibmissed; % .1
PPpenalty=dataPPpenalty*libPPpenalty;
if size(matchedmat) ≠ 0
    p_DPI=f_DPI(data,dataflag);
p_LPI=f_LPI(libisotope,libflag);
p_PP=f_PP(matchedmat);
p_AR=f_AR(matchedmat);
    out= p_DPI * p_LPI * p_PP * p_AR * PPpenalty;
else
    out=0;
end

A.2.7 makeOutput.m

% Output the identification results.
% Output model posteriors.

pv2=zeros(n_MacroModels,1);
modelposteriors={};
normalization=0;
for i=1:n_MacroModels
    current_posterior=max(pv(indexModelPosteriors(i).index)...
    );
normalization=normalization+current_posterior;
modelposteriors{i}={indexModelPosteriors(i).name};
\[ pv2(i) = \text{current posterior}; \]
\[ \text{end} \]
\[ pv2 = pv2 / \text{sum}(pv2); \]
\[ \text{[trash idx]} = \text{sort}(pv2, \text{'}descend\text{'}); \]
\[ \% \text{Print top model posteriors} \]
\[ \text{ModelPosteriorOutput} = \text{cell}(5, 4); \]
\[ \text{for } i = 1:5 \]
\[ \quad \text{ix} = \text{size} (\text{modelposteriors} \{\text{idx}(i)\} \{1\}, 1); \]
\[ \quad \text{for } j = 1:ix \]
\[ \quad \quad \text{ModelPosteriorOutput} \{i, j\} = \text{modelposteriors} \{\text{idx}(i) \ldots \} \{1\} \{j\}; \]
\[ \quad \text{end} \]
\[ \quad \text{ModelPosteriorOutput} \{i, 4\} = pv2(\text{idx}(i)); \]
\[ \text{end} \]
\[ \text{disp('Model Posteriors')} \]
\[ \text{ModelPosteriorOutput} \]
\[ \% \text{Output Isotope posteriors} \]
\[ \text{IsotopePosteriorOutput} = \text{cell}(n_{\text{uniqueIsos}}, 2); \]
\[ \text{pv3} = \text{zeros}(1, n_{\text{uniqueIsos}}); \]
\[ \text{for } i = 1:n_{\text{uniqueIsos}} \]
\[ \quad \text{indx} = \text{indexIsotopePosteriors}(i).\text{index}; \]
\[ \quad \text{IsotopePosteriorOutput} \{i, 1\} = \text{uniqueisotopes} \{i\} \{1\}; \]
\[ \quad \text{IsotopePosteriorOutput} \{i, 2\} = \text{sum}(pv2(\text{indx})); \]
\[ \quad \text{pv3}(i) = \text{IsotopePosteriorOutput} \{i, 2\}; \]
\[ \text{end} \]
\[ \% \text{IsotopePosteriorOutput} \]
\[ \% \text{Let's just output the top 5 isotope posteriors. To see ... everything,} \]
\[ \% \text{uncomment the above line.} \]
\[ \text{[trash idx]} = \text{sort}(pv3, \text{'}descend\text{'}); \]
\[ \text{for } i = 1:5 \]
\[ \quad \text{IsotopePosteriorOutput2} \{i, 1\} = \text{IsotopePosteriorOutput} \{\text{idx... (i), 1\};} \]
\[ \quad \text{IsotopePosteriorOutput2} \{i, 2\} = \text{IsotopePosteriorOutput} \{\text{idx... (i), 2\};} \]
\[ \text{end} \]
\[ \text{IsotopePosteriorOutput2, ModelPosteriorOutput} \]
A.3 Library/

This directory includes both an already generated library file (ModelLibrary_ANSIonly_Gauss.mat) and the code necessary to generate this library. The code presented is a simple version that is intended to run on a single machine. The pre-generated libraries were created on the Taub Campus cluster and took approximately one day to generate when using 20 nodes (12 cores each). It’s not really recommended to run this simple version to actually generate the full library due to runtime concerns, but it is included for completeness.

A.3.1 Library/makeIsotopeIndex.m

```matlab
1 % Create an index for outputting the individual isotope ... 
2 % posteriors.
3 % makeIsotopeIndex.m
4 % After computing the model posteriors, we will integrate ... 
5 % across them to
6 % find the posteriors for individual isotopes. This script... 
7 % creates an
8 % index to do this quickly.
9 %
10 %% Find how many unique isotopes are in the library.
11 n_uniqueIsos=0;
12 flag=0;
13 uniqueisotopes={};
14 for i=1:n_MacroModels
15     if size(ModelLibrary(i).name,1)==1
16         n_uniqueIsos=n_uniqueIsos+1;
17         uniqueisotopes{n_uniqueIsos}=ModelLibrary(i).name;
18         indexIsotopePosteriors(i).name=ModelLibrary(i).name...
19     end
20 else
21     break;
22 end
23 % Make the index.
24 % Initialize
```
A.3.2 Library/makeNameIndex.m

```matlab
% % makeNameIndex.m
% After computing the likelihood for each model, the ... likelihood vector must
% be processed. This script will create the a cell array ... that
%
7 nmodel=size(ModelLibrary,2);
8
9 % Figure out how many unique isotopes are in the library.
10 % The library is sorted such that all single isotope models...
11 % are first, then
```
% all two isotope models, and so on. Simply go through the...
list until the
% first multi-isotope model is found to find the number of ...
unique isotopes.

n_uniqueIsos=0;
flag=0;

for i=1:nmodel
   if size(ModelLibrary(i).name,1)==1
      n_uniqueIsos=n_uniqueIsos+1;
   else
      break;
   end
end

%% Create a key structure for the model posteriors
n_MacroModels= n_uniqueIsos + (n_uniqueIsos)*(n_uniqueIsos-1)/2 + (n_uniqueIsos)*(n_uniqueIsos-1)*(n_uniqueIsos-2)/6
indexModelPosteriors(n_MacroModels).name=0;
indexModelPosteriors(n_MacroModels).index=0;

% Set the first entry
counter=1;
lastmodel=ModelLibrary(counter).name;
indexModelPosteriors(counter).index= [1];
indexModelPosteriors(counter).name=ModelLibrary(1).name;

% Process the rest
for i=(counter+1):nmodel
   currentmodel=ModelLibrary(i).name;
   if isequal(currentmodel,lastmodel)
      indexModelPosteriors(counter).index = [...
         indexModelPosteriors(counter).index, i];
   else
      counter=counter+1;
      indexModelPosteriors(counter).name = ModelLibrary(...
         i).name;
      indexModelPosteriors(counter).index = [i];
   end
   lastmodel=currentmodel;
end
function [ out ] = reducemodel( model )

%REDUCEMODEL Reduces a model further.
% Eliminates peaks under 35 keV.

energies=model(:,1);
areas=model(:,2);
% uncs=model(:,3);

indx=find(energies > 35);

out=[energies(indx) areas(indx)];
end
A.3.4 Library/LibraryGeneration/simpleGenerateLibrary.m

```matlab
% Generate the model library.
%% simpleGenerateLibrary.m
% This script will create the library file needed for ...
% identification.
% This version is intended to run on a single local machine...
% instead of on
% the campus cluster. As such, it will take a long time to ...
% run (on the
% order of 20 days).
%
% 1. Generate a list of the activity ratios desired for the...
% library.
% These are stored as ratio vectors.
% Make all of the rat vectors
step=.1;
ratmax=1-step;
ratmin=step;
rat1=ratmin:step:ratmax;
nrat=(ratmax-ratmin)/step + 1 ;
% rat3=zeros(3,numb*(numb-1));
% two isos
rat2=[];
for i=1:nrat
    first=rat1(i);
    second=1-first;
    rat2=[rat2; first, second];
end
rat3=[];
step=.1;
firstv=step:step:(1-step);
% firstv=[.1 .2 .3 .4 .5 .6 .7 .8]
for i=1:size(firstv,2)
    first=firstv(i);
```
left=1-first;

for j=1:(left/step-.1)
    second=j*step;
    third=1-second-first;
    rat3=[rat3; first,second,third];
end
end

%% 2. Generate a pseudospectrum for each model.
% This assumes that there is a csv file for each isotope ...
% desired in this directory.
files=dir('Library/Library Generation/Isotopes/*.csv');

% Calibration.
cal.a=0;
cal.b=2.9287;
cal.c=1-cal.b*1;

nisos=0;
for file=files'
    nisos=nisos+1;
    name=file.name(1:end-4);
    csv=csvread(file.name);
    rawlib_ansig(nisos).name = name;
    rawlib_ansig(nisos).iso = csv;
    rawlib_ansig(nisos).cal = cal;
end
save('Library/Library Generation/rawlib_ansig','rawlib_ansig...');

% get calibration
cal.a=0;
cal.b=2.9287;
cal.c=1-cal.b*1;
ch=1:1024;
evec=ch*cal.b+cal.c;

% Initialize large structure file for all spectra
% Count how many elements this structure will have.
struct=0;
for i=1:nisos
    nstruct=nstruct+1;
end
for i=1:(nisos-1)
    for j=i:nisos
        nstruct=nstruct+1*size(rat2,1);
    end
end
for i=1:(nisos-2)
    for j=i:(nisos-1)
        for k=j:nisos
            nstruct=nstruct+1*size(rat3,1);
        end
    end
end

% Generate all spectra
% Initialize structure
ProblemSet_ansig(nstruct).iso = [];
ProblemSet_ansig(nstruct).rat = [];
ProblemSet_ansig(nstruct).spec = [];

% Single isotope models
nstruct=0;
for i=1:nisos
    nstruct=nstruct+1;
    ProblemSet_ansig(nstruct).iso= rawlib_ansig(i).name;
    ProblemSet_ansig(nstruct).rat= [1];
    y=GeneratePseudoSpectra2({rawlib_ansig(i).name},1,...
        rawlib_ansig);
    ProblemSet_ansig(nstruct).spec=y;
    ProblemSet_ansig(nstruct).spec= rawlib_ansig(i).iso;
end

% double isotope models
for i=1:(nisos-1)
    for j=(i+1):nisos
        for r=1:size(rat2,1)
nstruct=nstruct+1;

ProblemSet_ansig(nstruct).iso= {rawlib_ansig(i)... 
    .name, rawlib_ansig(j).name};
ProblemSet_ansig(nstruct).rat= rat2(r,:);

y=GeneratePseudoSpectra2(ProblemSet_ansig(... 
    nstruct).iso, rat2(r,:), rawlib_ansig);
ProblemSet_ansig(nstruct).spec=y;

% ProblemSet_ansig(nstruct).spec= rawlib_ansig(... 
    i).iso*rat2(r,1)+rawlib_ansig(j).iso*rat2(r,2);
end
end
end
end

% triple isotope models
for i=1:(nisos-2)
    for j=(i+1):(nisos-1)
        for k=(j+1):nisos
            for r=1:size(rat3,1)
                nstruct=nstruct+1;

                ProblemSet_ansig(nstruct).iso= {... 
                    rawlib_ansig(i).name; rawlib_ansig(j)... 
                    .name;rawlib_ansig(k).name};
ProblemSet_ansig(nstruct).rat= rat3(r,:);

                y=GeneratePseudoSpectra2(ProblemSet_ansig(... 
                    nstruct).iso, rat3(r,:), rawlib_ansig);
ProblemSet_ansig(nstruct).spec=y;
            end
        end
    end
end
save('Library/LibraryGeneration/ProblemSet_ansig', ...
    ProblemSet_ansig')

%% 3. Extract peak information.
    cal.a=0;
    cal.b=2.9287;
    cal.c=1-cal.b*1;
ModelLibrary(nstruct).lib=0; % Initialize structure
for iproblem=1:nstruct
    y=ProblemSet_ansig(iproblem).spec;
    results=WaveletCodeQuiet(y,cal);
    ModelLibrary(iproblem).lib= [results.centroid,...
        results.area,results.unc ./ results.area];
    ModelLibrary(iproblem).name=ProblemSet_ansig(iproblem)... iso;
    ModelLibrary(iproblem).rat= ProblemSet_ansig(iproblem)... rat;
end
% save('Library/demoModelLibrary','ModelLibrary')