METHODOLOGY FOR ANOMALOUS SOURCE DETECTION IN SPARSE GAMMA-RAY SPECTRA

BY

GREGORY R. ROMANCHEK

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Masters of Science in Nuclear, Plasma, and Radiological Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Master's Committee:

Professor Shiva Abbaszadeh, Adviser
Professor Angela Di Fulvio

ABSTRACT

The dangers of rogue nuclear material remain a top concern despite increased attention and strides in computational protocols. Single, mobile detector methodologies for localizing sources via autonomous surveying have become popular with the maturation of the machine learning (ML) and statistical learning (SL) fields as well as increased access to drone (quad-copter) technology. These options, however, face task-inherent impediments which either degrade the quality of collected gamma-ray spectra or necessitate high-quality information on source and background spectrum compositions. Some such hurdles include: limited dwell periods, fluctuating and/or unknown background, weak source signal due to large distance and/or small/shielded activity, and the low sensitivity of mobile detectors. As such, collected gamma-ray spectra are sparse, containing many zero-count energy channels, and contain relatively large background presence. This combination of factors, as well as the natural variance in second-to-second count rates, leads to low-quality information for making navigational decisions. In this thesis, an SL algorithm is presented for extracting source count estimations from time-series, sparse gamma-ray spectra with no prior training required. A Gaussian process with a linear innovation sequences procedure is used to efficiently update ongoing spectral estimates with real-time training and hyperparameters defined by detector characteristics. Being free of prior training and assumptions allows such an algorithm to be used in a wide variety of sparse-data settings whereas a trained solution would have very narrow applications. We have evaluated the effectiveness of this approach for anomaly detection using background spectra dataset collected with a Kromek D3S and simulated source spectra. Results of anomaly detection testing with a source count rate at half that of the background displays an area under the ROC curve of 0.9. Further, deployment with an ML guided navigation scheme shows, after an anomaly is detected, estimated gross source counts and true gross source counts have an average correlation of 0.998, whereas estimated gross background counts and true gross background counts have an average correlation of 0.876.

ACKNOWLEDGMENTS

I would like to express tremendous thanks and gratitude to my advisor, Professor Shiva Abbaszadeh, who offered continuous support and encouragement throughout my Master's program. I have had the fortune of being present at the start of her Radiological Instrumentation Laboratory (RIL) family, the marvel of her dedication and perseverance with managing RIL and her baby, and the honor of remaining an always valued and trusted student through her career growth. I look forward to the remainder of my academic career with her.

I would also like to express my sincerest thanks to my thesis committee member, Professor Angela Di Fulvio, for her insightful comments and feedback. I would like to acknowledge the Consortium for Nonproliferation Enabling Capabilities (CNEC) for providing funding support for much of this work and the professional network and development they offered. Lastly, I would like to acknowledge my family, and parents in particular, for their support, trust, and love.

To my wife.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 PROBLEM DESCRIPTION

Advanced source surveying protocols for localizing rogue nuclear material are crucial tools in the various national security scenarios for which the presence and location of radioactive material must be quickly determined. However, several factors inherent to the source surveying task degrade or otherwise limit the quality of information acquired during surveying. This information is typically gamma-ray counts recorded by a gamma-ray detector, presented in either counts per second (cnts/s) or in the form of a gamma ray spectrum. In both cases, source presence signatures – be it elevated count rates or identifiable source photopeaks – are used to make inferences about source location, strength, and identity.

Focusing on surveying with a single mobile detector, whether using a handheld detector and manually surveying or using advanced source localization algorithms with detectors mounted on drones, many problematic similarities exist. Dwell times are often very short (on the order of seconds) [1-3] leading to low count statistics. There is also a potential for a large distance between the source and the detector, and the possibility that no source is present at all. Additionally, naturally occurring radioactive material (NORM) is ubiquitous, leading to background/noise gamma-rays obfuscating the source signal. Because of these factors and the low sensitivity of handheld/mountable detectors, collected spectra will usually be background dominated and contain very few counts. In these cases, machine learning (ML) and statistical learning (SL) surveying algorithms can falter due to lack of relevant information (source counts) compared to the presence of noise (background counts). Because of this, such protocols often default to predefined strategies like a uniform search method or random walk until greater source information

is present [1, 2]. Thus, the benefits of advanced surveying techniques cannot be realized until they leave this period of unoptimized pathing.

Providing high quality source information is necessary for optimizing the performance of ML and SL surveying protocols. Due to the hurdles native to this task, it is not always guaranteed that such information is available, leading to decreased performance in a high-stakes scenario. Additionally, in surveying scenarios, the identity of the source and background distribution should be assumed unknown for the most generalizable solution. This eliminates the potential for learned algorithms due to the immense variety of encounterable source and background combinations, strengths, and natural randomness. Finally, methodologies need to be computationally efficient such that they can perform real-time analysis of incoming data. This is necessary if the output is being directly fed to a navigational algorithm, and is crucial for the speed any localization task requires.

1.2 GAMMA-RAY RADIATION AND SPECTRA

Gamma-ray radiation is the radiative emission of a high-energy photon – usually ranging from a few keV to several MeV – following a form of nuclear decay (found commonly in nature), nuclear reaction (found in particle accelerators and nuclear reactors), annihilation reaction between matter and anti-matter (found in PET imaging), and more [4]. Gamma-ray generation is not restricted to the above mechanisms or source examples, but rather result in general from nuclear re-arrangement, whereas X-ray, lower energy photons, originate from electron interactions. Instead, it is sufficient to understand: 1) where gamma-rays generally come from, and 2) the finger-print like quality of gamma-rays in spectroscopy.

Addressing point one, gamma-rays are high-energy photons which originate from a series of natural and artificial physical interactions and mechanisms. In source surveying, we are concerned with gamma-ray linked to elements and cosmic rays. Terrestrial sources are often linked to specific radioactive material which undergoes decay more readily than stable material. The energy of these emitted gammas is constant with the elemental source and the specific mechanism taking place. Taking Beta-decay as an example, the decay scheme for ^{137}Cs can be seen in Figure 1.1 which results in the emission of a 662 keV gamma-ray from the meta-stable ^{137m}Ba in 85.1% of decays. This isotope is a common fission product of ^{235}U, and is thus important for its presence in background and in security applications. NORM is an omnipresent and varied source of gamma-rays depending on geography and climate. Gammas belonging to the NORMs of ^{40}K, daughters of ^{238}U, ^{232}Th, ^{235}U, and more dominate the background energy spectrum from 0 to 3 MeV [5-7]. Cosmic rays, like NORMs, are ubiquitous and varied, contributing to background noise.

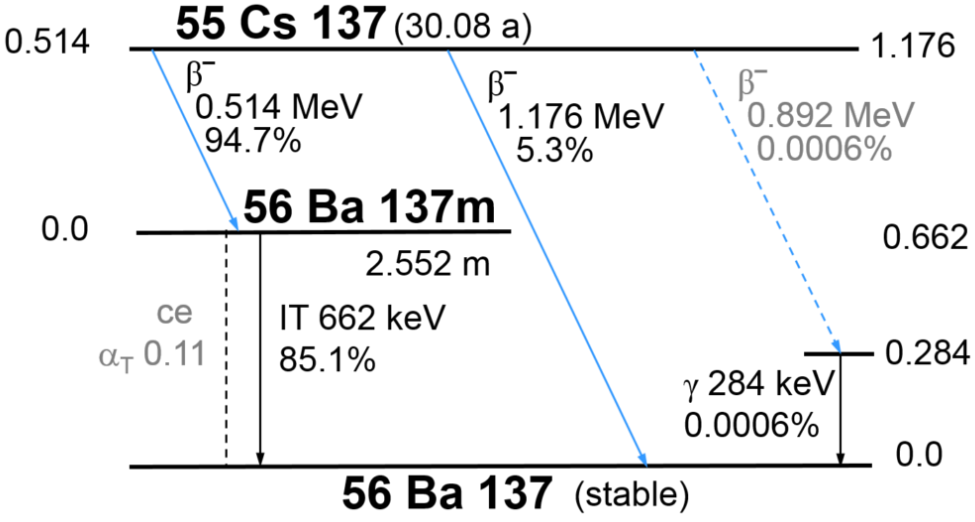

Figure 1.1: ^{137}Cs decay scheme [8].

Addressing point two, each elemental source, described by the number of protons, neutrons, and excitation state, has specific gamma-ray energies associated with it. A gamma-ray spectrum results from a gamma-ray spectrometer detecting incident gamma-rays and binning them into discrete energy levels. An example of a high-count spectrum with labelled features can be seen in Figure 1.2 where the x-axis is the energy level of the detected photon and the y-axis is the frequency of counts. ^{24}Na has two primary gamma-ray emission energies, at 1369 keV and 2754 keV. As these photons interact with the detector, some smearing occurs as defined by the detector's energy resolution at that specific energy. This leads to peaks forming about specific energy channels; these are known as photopeaks. As photopeaks form, evidence for the presence of specific sources grows. Figure 1.2 illustrates this concept as the two photopeaks from ^{24}Na encompasses a much larger range than a single channel. Background sources, electronic noise, and additional physical interactions within the detector similarly contribute to spectrum counts and are also subject to the smearing due to limited energy resolution and additional effects.

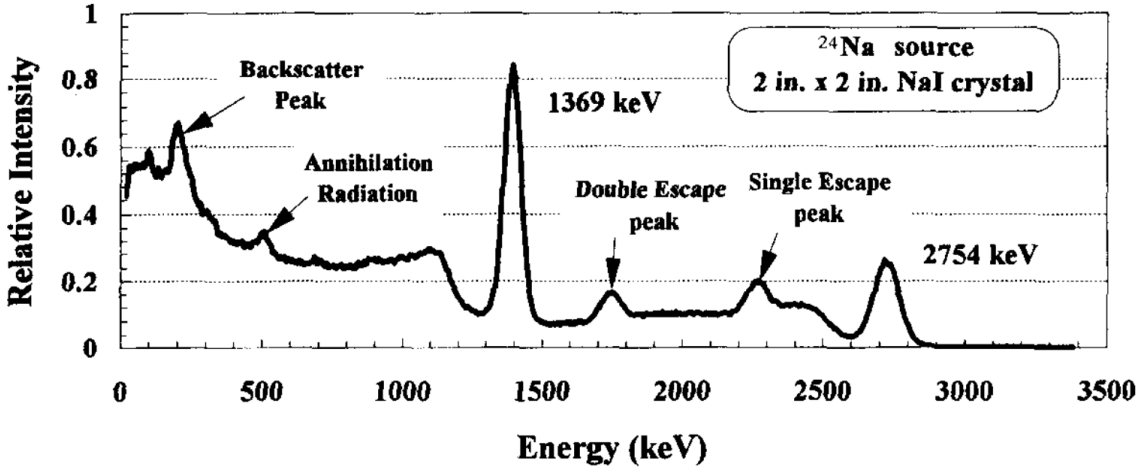

Figure 1.2: Developed gamma-ray spectrum [4].

1.3 RELATED WORKS

In order to address the problem at hand, the quality of source information must be improved by either increasing source presence or reducing background presence in the spectrum.

While there are computational methods for background estimation, many fail to meet the established criteria for source localization problems as they require either defined peak regions [8r], the tuning of parameters [10], or rely on fully-developed (high count) spectra [11]. These are unsatisfactory for source-search problems as the specific source identity may not be known, and thus its photopeak region cannot be selected. Secondly, manually tuning or training parameters requires knowledge of the given system/algorithm, predictable data, and time, none of which may be present in emergency situations. Finally, given extremely sparse spectra, many of the provided techniques of background removal simply do not function as they rely upon features within high-count spectra.

Alamoniotis et al. [12], however, developed a kernel-based Gaussian process (GP) for background estimation under sparse conditions and with no prior information. Their quantification of "sparse," however, (~435 background counts/s achieved with a 3 in × 3 in NaI detector) is more than four times that achieved with handheld detectors such as the Kromek D3S [13] over the same timescale which are capable of being hoisted with drones. Thus, the Kromek D3S, which has already been deployed in mobile sensor networks [14-16] provides a better representation of the background count rate (~50 counts/s) for the mobile detectors likely to be used in source surveying scenarios. This definition of "sparse" is a necessary consequence of mobile source surveying, and so a method for analyzing this level of sparse spectra is needed.

Alternative machine learning algorithms exists which have the potential to estimate source spectra from collected spectra through both regression and categorical approaches, such as the

increasingly popular neural networks. With the conditions set out previously, however, such techniques are not well suited for this application. Namely, any machine learning methodology with substantial training requirements would hinder the generalizability of the algorithm. With substantial variation in background activity and composition, diverse combinations of source type and activity, and biases inherent to the specific detector type used, too many variables come into play to reasonably incorporate sufficient training examples to learn all cases accurately. In addition to the breadth of encounterable spectra, time dependencies need to be taken into account, adding a dimension of complexity to the training data required.

1.4 PROPOSED SOLUTION

The unique feature of source localization protocols acquiring an independent spectrum sample at each new testing location is taken advantage of in the proposed solution. Since a number of gamma spectra are collected over the total surveying period, one can view these collections as a sequence of noisy samples of the true source spectrum distribution. For each individual spectrum collected, source data may be insignificant; but using all the samplings collectively can improve the confidence of estimated source presence. Combining this with sparse data background removal provides a statistical framework for estimating source presence in surveying protocols.

In this thesis, we present a statistical algorithm for estimating source presence under the sparse conditions inherent to source surveying protocols. Background presence is first reduced in collected spectra by using a GP structure presented in [12] but optimized for the sparse data at hand. Each newly acquired background-removed collection is then used to update an ongoing source distribution estimate via a Linear Innovation Sequences (LIS) scheme. For this algorithm, also presented in *Romanchek et al.* [29], we assume no prior information on background or source,

and the source distribution is estimated purely based on the readings acquired while surveying. GP and LIS were selected for both background removal and source distribution prediction as they require no previous training and are computationally inexpensive, allowing the algorithm to be executed by on-board equipment in real-time.

1.5 CHAPTER OVERVIEW

Chapter 1 covered the problem description, background, and an overview of the proposed solution. The four remaining chapters will walk the reader through the mathematical framework, algorithm structure, and validation procedures. Chapter 2 details the statistics behind GPs and LIS in context with the problem at hand. Chapter 3 uses the formalisms defined in Chapter 2 to provide a description of the algorithm used to reduce background presence, generate a source estimate, and conduct anomaly detection. Chapter 4 provides details on the validation methods and a description of the data sets used. Chapter 5 presents the results of the experiments outlined in the previous chapter. Chapter 6 concludes the thesis and provides future work regarding the proposed methodology.

CHAPTER 2: THEORY

2.1 GUASSIAN PROCESSES IN CONTEXT

When attempting to estimate the form and strength of the relationship f between a dependent variable (or observation) y and independent variable(s) $\boldsymbol{x}$ (or features), given by $y = f(\boldsymbol{x})$, regression is among the most common methodological paradigms. In a typical regression approach, the statistical estimation of this relationship is accomplished by constraining f to a class of test functions, then selecting function parameters via one of many optimization schemes over training data. A downside of this approach is that only function parameters are optimized and not the test function selection itself, leaving the user to manually chose it and leaving potentially more accurate solutions on the table. Gaussian process (GP) regressions, on the other hand, offer a means to more-or-less optimize the function selection itself. This is done in a theoretical sense by considering all possible test functions simultaneously and defining a prior probability distribution over this set such that higher probabilities dictate a greater likelihood that the function represents the data. This approach is more flexible, as it does not constrain the estimation to certain classes of functions [18, 19]. To understand GP regression, first, consider the general regression set-up:

$$y = \sum_{m=1}^{M} w_m \phi_m(\boldsymbol{x}) \tag{2.1}$$

Here, y is the scalar output representing the observations, $\boldsymbol{x} = \{x_1, \dots, x_N\}$ is an N-dimesnional input vector representing the features, ϕ_m are the basis functions representing a mapping from feature space to a transformed space defining $\boldsymbol{\phi} = \{\phi_1(\boldsymbol{x}), \dots, \phi_M(\boldsymbol{x})\}$, w_m are the associated scalar weights defining $\boldsymbol{w} = \{w_1, \dots, w_M\}$, and M is the number of basis functions considered. As an example, the common linear regression $y = mx + b$ can be derived from Equation 2.1 by considering $\boldsymbol{x} = \{x\}$ (single feature) and $\boldsymbol{\phi} = \{\phi_1(x), \phi_2(x)\}$, where $\phi_1(\boldsymbol{x}) = mx$ and $\phi_2(\boldsymbol{x}) =$

b. Typically, we are given D training samples with: $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(D)}$ where $\boldsymbol{x}^{(d)} = \{x_1, \dots, x_N\}$, and corresponding observations $\boldsymbol{y} = \{y_1, \dots, y_D\}$. Thus, training sample 1 is $\{\boldsymbol{x}^{(1)}, y_1\}$. In the context of our problem, we consider a single input feature, the energy channel, $\boldsymbol{x}^{(d)} = \{x_d\}$ and the corresponding observation y_d of counts in this channel. One full spectrum consists of observations $\{x_d, y_d\}$ for $d = \{1, \dots, D\}$ where D is the number of detector channels. In this way, we can envision the spectrum as a function for counts varying with channel number. In a similar sense, we can take observation pairs to be channel number and background or source counts, instead of total counts, in a given channel. Equation 2.1 can be represented in vector form as:

$$\boldsymbol{y} = \boldsymbol{\Phi w} \tag{2.2}$$

where $\boldsymbol{\Phi}$ is the $D \times M$ design matrix whose elements are $\Phi_{(d,m)} = \phi_m(\boldsymbol{x}_d)$ where $m = \{1, \dots, M\}$ and $d = \{1, \dots, D\}$. GPs allow us to avoid defining the basis functions, and so the final estimate will be independent of M.

From here, we move toward deriving the GP estimator by casting the regression problem into a Bayesian formalism. A normal prior distribution over the weight vector takes the form:

$$P(\boldsymbol{w}) = N(\mathbf{0}, \sigma_w^2 \mathbf{I}) \tag{2.3}$$

The mean of which is zero, and each weight has uniform variance σ_w^2. This implies the weights are uncorrelated and that their distribution is governed by the hyperparameter σ_w^2. It is common to assume this prior distribution for the weights due to the lack of prior information [2, 3]. Since Equation 2.2 is now defined as a linear combination of jointly Gaussian variables, $\boldsymbol{y}$ itself is Gaussian. Thus, its expectation value and variance are [18]:

$$E[\boldsymbol{y}] = E[\boldsymbol{\Phi w}] = \boldsymbol{\Phi} E[\boldsymbol{w}] = 0 \tag{2.4}$$

$$\begin{aligned} cov(\boldsymbol{y}) &= E[(\boldsymbol{y} - E[\boldsymbol{y}])(\boldsymbol{y} - E[\boldsymbol{y}])^{\mathrm{T}}] \\ &= E[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}] = \boldsymbol{\Phi} E[\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}] \boldsymbol{\Phi}^{\mathrm{T}} = \sigma_w^2 \boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}} = \mathbf{K} \end{aligned} \tag{2.5}$$

where $\mathbf{K}$ is the $D \times D$ Gram Matrix with elements:

$$\mathbf{K}_{(i,j)} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_w^2 \boldsymbol{\phi}(\boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_j) \tag{2.6}$$

where $i, j = \{1, \dots, D\}$ and $k(\boldsymbol{x}, \boldsymbol{x}')$ is the kernel function with scalar output. Kernel representation condenses the inner product in feature space without needing to explicitly define the feature mapping $\boldsymbol{\phi}$ itself. This inner product, the kernel function, is the covariance between $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the transformed space and is discussed further in section 2.2 [18]. The prior over our output vector $\boldsymbol{y}$ then follows the distribution:

$$P(\boldsymbol{y}) = N(\mathbf{0}, \mathbf{K}) \tag{2.7}$$

In random processes, however, there is often assumed to be some noise within the observations of the target values such that the true observation is:

$$t_d = y_d + \epsilon_d \tag{2.8}$$

Here, $\boldsymbol{\epsilon}$ represents the noise in observations $\boldsymbol{y}$, where each element ϵ_d are assumed to be uncorrelated with each other and normally distributed with mean zero and variance σ_d^2. Physically, for an observation of counts y in channel x, y is a sampling of a random variable that can take a range of values due to the random nature of radiation and energy resolution of the detector. The prior over $\boldsymbol{t} = \{t_1, \dots, t_D\}$ is subsequently [18]:

$$P(\boldsymbol{t}) = N(\mathbf{0}, \mathbf{K} + \sigma_d^2 \mathbf{I}) \tag{2.9}$$

Since the goal of GP is regression, we need to predict new values in $\boldsymbol{t}$ given features $\boldsymbol{x}$. Suppose you are given the output vector $\boldsymbol{t} = \{t_1, \dots, t_D\}$ and corresponding input vector $\boldsymbol{x} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_D\}$.We would like to estimate point t_{D+1} for a new input $\boldsymbol{x}_{D+1}$; i.e., we want to compute the probability distribution for t_{D+1}, $P(t_{D+1}|\boldsymbol{t})$, given training set $\{\boldsymbol{x}, \boldsymbol{t}\}$ and testing point $\boldsymbol{x}_{D+1}$.

From this probability distribution, we can acquire an estimated mean and variance for t_{D+1}. By conditional probability

$$P(t_{D+1}|\boldsymbol{t}) = \frac{P(\boldsymbol{t},t_{D+1})}{P(\boldsymbol{t})} \tag{2.10}$$

Thus, computing $P(\boldsymbol{t}_{D+1}) = P(t_1, \dots, t_D, t_{D+1})$ is necessary. As $t_1, \dots, t_D, t_{D+1}$ are jointly Gaussian, $\boldsymbol{t}_{D+1}$ is Gaussian as well with distribution [18]:

$$P(\boldsymbol{t}_{d+1}) = N\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma_d^2\mathbf{I} & \mathbf{k} \\ \mathbf{k}^{\mathrm{T}} & k(\boldsymbol{x}_{d+1}, \boldsymbol{x}_{d+1}) \end{bmatrix}\right) \tag{2.11}$$

where the elements of $\mathbf{k}$ are given by $\mathrm{k}_{(i)} = k(\boldsymbol{x}_i, \boldsymbol{x}_{D+1})$ for $i = \{1, \dots, D\}$. Then, the conditional distribution in Equation 2.10 is the conditional distribution of two Gaussian functions. The mean and covariance of Equation 2.10 are then [18]:

$$\mu_{t_{D+1}|\boldsymbol{t}} = \mathbf{k}^{\mathrm{T}}\,(\mathbf{K} + \sigma_d^2\mathbf{I})^{-\mathbf{1}}\mathbf{t} \tag{2.12}$$

$$\Sigma_{t_{D+1}|\boldsymbol{t}} = k(\boldsymbol{x}_{D+1}, \boldsymbol{x}_{D+1}) - \mathbf{k}^{\mathrm{T}}\,(\mathbf{K} + \sigma_d^2\mathbf{I})^{-1}\,\mathbf{k} \tag{2.13}$$

Here, Equations 2.12 and 2.13 are used to predict the mean and variance of t_{D+1} given input $\boldsymbol{x}_{D+1}$, kernel function k, and training data. For predicting a series of new points $\boldsymbol{t}^*$ with features $\boldsymbol{x}^*$, the scalar kernel function k is replaced with the covariance matrix of $\boldsymbol{t}^*$, and $\mathbf{k}$ becomes the $D \times D^*$ covariance matrix between $\boldsymbol{x}$ and $\boldsymbol{x}^*$, where D^* is the number of points in $\boldsymbol{t}^*$.

If the observations are known to have non-zero mean – as we expect with radiation counts –, the simple transformation of $\boldsymbol{t}' = \boldsymbol{t} - \mathrm{E}[\boldsymbol{t}]$ creates a new random variable $\boldsymbol{t}'$ with mean zero, and the GP estimator derivation proceeds identically, ultimately yielding:

$$\boldsymbol{\mu}_{\boldsymbol{t}^*|\boldsymbol{t}} = \boldsymbol{\mu}_{\boldsymbol{t}^*} + \mathbf{k}^{\mathrm{T}}\,(\mathbf{K} + \sigma_d^2\mathbf{I})^{-\mathbf{1}}(\mathbf{t} - \boldsymbol{\mu}_{\mathbf{t}}) \tag{2.14}$$

The covariance is the same as in Equation 2.13 but with $\boldsymbol{x}_{D+1}$ replaced with $\boldsymbol{x}^*$. Now, we can predict the mean and variance of evert test point in $\boldsymbol{x}^*$ given a set of training data $(\boldsymbol{x}, \boldsymbol{t})$. Traditionally, the mean is used as the prediction where the variance of each prediction is used as

a pseudo-confidence measure. Having now Equations 2.13 and 2.14 to make prediction, we are left with the need to choose a kernel function k which best captures the variance in the data sets and the need to properly define our testing and training data. An overview of kernel functions is in Section 2.2 while implementation is in Section 3.2.

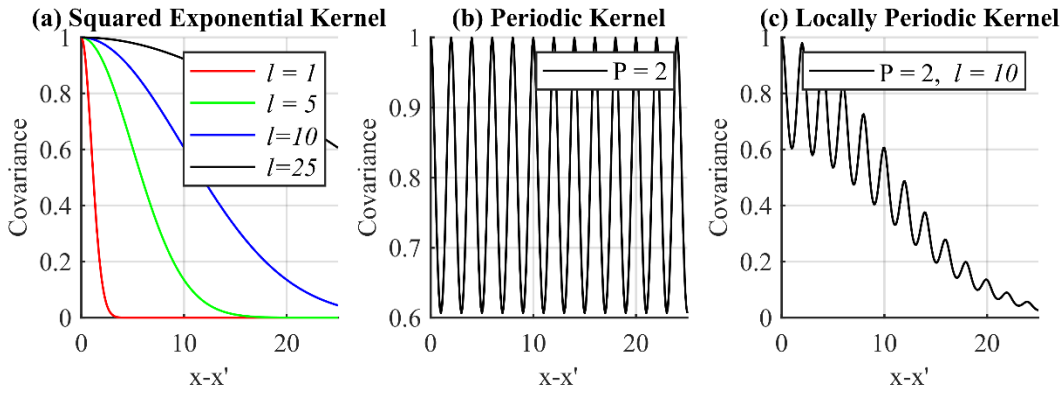

Figure 2.1: Examples of the Squared Exponential (a), periodic (b), and locally periodic (c) kernels.

2.2 KERNEL FUNCTIONS IN GAUSSIAN PROCESSES

Kernel functions define the covariance between x and x' in a feature mapped space without needing to explicitly define the feature transformation $\phi(x)$ [18]. As a note, kernel functions can have vector or scalar input. The notation in this section uses scalar inputs for clarity. Equation 2.6 provides the definition of a general kernel function, though it does not provide the mathematical structure as the basis functions are undefined. As an example, perhaps the simplest kernel is the linear kernel, in which $\phi(x) = x$ such that the kernel function is:

$$k(x, x') = xx' \tag{2.15}$$

Not all functions are valid kernel functions, however. The primary condition to satisfy is that the Gram Matrix constructed from a given kernel must be positive semidefinite for all possible

choices of x [18]. Since the purpose of a kernel-based approach in GPs is to conveniently define covariance, one can simply select a kernel which is known to be valid and captures the covariance in a desired way. One popular kernel is the Gaussian (or squared exponential) kernel (Figure 2.1a) [19]:

$$k_G(x, x') = \sigma^2 \exp\left(-\frac{(x-x')^2}{\ell^2}\right) \tag{2.16}$$

where ℓ is the characteristic length controlling how quickly covariance decays between similar inputs, and σ^2 is the output variance, analogous to the weight variance in Equation 2.6. The Gaussian kernel assets that the covariance between inputs is Gaussian such that covariance is highest between two of the same inputs and decays as the inputs increase in "distance". A structurally different kernel is the periodic kernel (Figure 2.1b) [19]:

$$k_P(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2\left(\frac{\pi|x-x'|}{p}\right)}{\ell^2}\right) \tag{2.17}$$

where p defines the periodicity of the covariance, and σ and ℓ are the same as in Equation 2.16. Such a kernel asserts that the covariance oscillates periodically between high values and low values as distance between inputs change. In these kernels, ℓ, σ, and p are examples of hyperparameters which need to be selected or optimized for over a given data set, to be discussed in section 2.3.

Valid kernels can also be combined or transformed to form new kernels to capture unique features. Given valid kernels $k_1(x, x')$ and $k_2(x, x')$, these can be transformed into a new valid kernel $k_3(x, x')$ following a selection of permitted transformations and combinations such as:

$$k_3(x, x') = c\, k_1(x, x') \tag{2.18}$$

$$k_3(x, x') = k_1(x, x') + k_2(x, x') \tag{2.19}$$

$$k_3(x, x') = k_1(x, x') \times k_2(x, x') \tag{2.20}$$

Where c is a constant. Given Equation 2.20, the periodic and Gaussian kernels can be multiplied to form the locally periodic kernel (Figure 2.1c), where $r = x - x'$:

$$k_{LP}(r) = \sigma^2 \exp\left(-\frac{1}{\ell^2}\left(d^2 + 2\sin^2\left(\frac{\pi|r|}{p}\right)\right)\right) \tag{2.21}$$

This kernel has a macro feature of Gaussian covariance, decaying with more distant inputs, as well as a micro feature of oscillations along that decay.

Selection of a kernel function and its hyperparameters depends upon the data and is ultimately what determines the effectiveness and generalizability of the GP used. While methods for automatically selecting kernels in some machine learning methods such as Support Vector Machines (SVMs) exist [20], the predominant method for choosing kernels in GP regression is by selecting a finite number of kernels which may provide good results given your data, testing each, and selecting from among them [21]. This inference approach requires the analyst know something about the data since an infinite number of kernels exits.

The primary distinction in classes of kernel functions are stationary vs. nonstationary [18]. Stationary processes satisfy $k(x, x') = k(x - x')$ whereas nonstationary do not. This implies stationary processes are independent of shifts in input depending solely on the distance between inputs. Nonstationary processes are dependent upon input values rather than the distance between them. The aforementioned Gaussian kernel is a stationary kernel whereas the linear kernel is nonstationary. Smoothness, how well a kernel handles discontinuity, is another distinction in class. Stationary kernels should be selected for stationary processes, and smooth kernels should be selected for smooth data.

In the problem at hand, sparse spectrum data varies abruptly and discretely between neighboring channels, making it very nonsmooth; but the overall distribution of source presence

per channel is expected to vary relatively smoothly. In both cases, the process is stationary as distance between the channels dictates how much information we expect them to share.

2.3 HYPERPARAMETER SELECTION

Kernel functions typically contain some number of hyperparameters which need to be selected prior to use. These include the previously mentioned characteristic length ℓ and periodicity p. Hyperparameter selection can be as influential as the kernel itself as seen in Figure 2.1a. Typically, GPs go through a training phase where the hyperparameters are selected to optimize the fit over the training data before predictions are computed. This is done by optimizing $P(\boldsymbol{t}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the hyperparameter vector, for an optimal hyperparameter selection $\widehat{\boldsymbol{\theta}}$. This requires reliable training data and the assumption that the generalizable best choice of hyperparameters is computable from the data at hand. While optimization strategies for computing $\widehat{\boldsymbol{\theta}}$ exist, we omit these in favor of fixed, system-defined hyperparameters. This choice is motivated by the context of the problem and the time constraints it enforces. Namely, we can assert that the covariance between detector channels has a fixed structure with physically relevant hyperparameters. This allows/requires us to maintain the same hyperparameters from spectrum to spectrum and enables us to omit hyperparameter the training phase.

2.4 LINEAR INNOVATION SEQUENCES IN CONTEXT

The GP regression allows us to construct a coarse prediction of source or background counts in each channel assuming we have proper training data. This prediction, however, will change from spectrum to spectrum. Linear Innovation Sequences lets us use these successive spectrum predictions to construct a more robust, updating background/source spectrum estimate. Let us

assume we have a series of spectra consisting of the estimated source counts in each channel and our goal is to estimate a single, smooth source spectrum. This source estimation task has a sequence of inputs $\{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(T)}\}$, where $\boldsymbol{y}^{(t)} = \{y_1^{(t)}, \ldots, y_D^{(t)}\}$; $y_d^{(t)}$ is the estimated source counts in channel d for the t^{th} spectrum; T is the number of spectra; and D is the number of detector channels. The goal is to estimate the source presence $\boldsymbol{z} = \{z_1, \ldots, z_D\}$ in each channel after the T^{th} newest collection given by $\boldsymbol{s}^{(T)} = \left\{s_1^{(T)}, \ldots, s_D^{(T)}\right\} = \mathrm{E}[\boldsymbol{z}|\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(T)}]$. If we were to use all this data at once, in one large estimation process, then it has a training set of size $T * D$ and testing set of size D. The covariance matrix **K** in Equation 2.14 is then of size $(T * D) \times (T * D)$. For example, for a detector with 1024 channels, by the tenth collection, the covariance matrix will be of size 10240×10240.

The size of the training data is problematic since the primary complexity cost of GPs is the inversion of the covariance matrix in Equation 2.14. While methods for inverting large matrices quickly (in fewer computational steps) exist, such as Cholesky Decomposition [22], the increasing computational cost is unavoidable, leading to slow prediction times. An alternative to inverting an ever-increasing covariance matrix is needed. Such computational scaling issues inherent to classic GP implementation have been alleviated via sparse GP approximations [23-25], but such approaches introduce additional complexities including the need to initialize large sparse matrices or to utilize batches of training samples. These drawbacks are not of consequence under normal computational conditions, but here data enters the algorithmic pipeline serially in time and has no upper time bound (more time is more spectra to analyze). Additionally, since we fully omit hyperparameter training, the benefits these advanced approaches offer are reduced.

Linear Innovative Sequences (LIS) provide a solution to this inversion problem, requiring the inversion of only the most recently acquired sequence to update an existing estimate. Namely,

let $\boldsymbol{z}, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(T)}$ be random vectors with finite second moments. First, Linear Minimum Mean Squared Error (LMMSE) estimation provides a solution for the estimation of $\mathrm{E}[\boldsymbol{z}|\boldsymbol{y}^{(t)}]$ as [26]:

$$\widehat{\mathrm{E}}\left[\boldsymbol{z}\middle|\boldsymbol{y}^{(t)}\right] = \mathrm{E}[\boldsymbol{z}] + \mathrm{Cov}\left(\boldsymbol{z}, \boldsymbol{y}^{(t)}\right)\mathrm{Cov}\left(\boldsymbol{y}^{(t)}\right)^{-1}\boldsymbol{y}^{(t)} \tag{2.22}$$

for $\boldsymbol{y}^{(t)}$ with zero mean. For nonzero mean $\boldsymbol{y}^{(t)}$, one can substitute these with $\boldsymbol{y}^{(t)\prime} = \boldsymbol{y}^{(t)} - \mathrm{E}[\boldsymbol{y}^{(t)}]$. Note the similarity in form between Equations 2.22 and 2.14. In fact, if $\boldsymbol{z}$ is taken as test points $\boldsymbol{t}^*$ and $\boldsymbol{y}^{(t)}$ as training points $\boldsymbol{t}$, then these estimators are identical except for their covariance definition. Explicitly, Equation 2.14 relies upon a kernel definition while Equation 2.22 uses the traditional covariance definition. Second, for $\boldsymbol{y}^{(t)}$ which satisfies $E\left[\boldsymbol{y}^{(t)}\right] = 0$ and $E\left[\boldsymbol{y}^{(i)}\boldsymbol{y}^{(j)\mathrm{T}}\right] = 0$ for $i \neq j$ (orthogonality condition), it can be shown that [26]:

$$\boldsymbol{s}^{(T)} = \widehat{\mathrm{E}}[\boldsymbol{z}|\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(T)}] = \mathrm{E}[\boldsymbol{z}] + \sum_{t=1}^{T}\widehat{\mathrm{E}}\left[\boldsymbol{z} - \mathrm{E}[\boldsymbol{z}]\middle|\boldsymbol{y}^{(t)}\right] \tag{2.23}$$

This provides the solution to our desired estimation. However, we do not know $\mathrm{E}[\boldsymbol{z}]$ – the expected source counts in every channel – by definition of our problem. Neither can we say that $E\left[\boldsymbol{y}^{(t)}\right] = 0$ or assert they are orthogonal outright. LIS, fortunately, provides a direct solution for imposing $E\left[\boldsymbol{y}^{(t)}\right] = 0$ and orthogonality via the transformation [26]:

$$\widetilde{\boldsymbol{y}}^{(t)} = \boldsymbol{y}^{(t)} - \widehat{\mathrm{E}}\left[\boldsymbol{y}^{(t)}\middle|\boldsymbol{Y}^{(t-1)}\right] \tag{2.24}$$

where $\boldsymbol{Y}^{(t-1)} = \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(t-1)}\}$ and:

$$\widehat{\mathrm{E}}\left[\boldsymbol{y}^{(t)}\middle|\boldsymbol{Y}^{(t-1)}\right] = \mathrm{E}\left[\boldsymbol{y}^{(t)}\right] + \sum_{i=1}^{t-1}\mathrm{Cov}\left(\boldsymbol{y}^{(t)}, \widetilde{\boldsymbol{y}}^{(i)}\right)\mathrm{Cov}\left(\widetilde{\boldsymbol{y}}^{(i)}\right)^{-1}\widetilde{\boldsymbol{y}}^{(i)} \tag{2.25}$$

such that [26]:

$$\widetilde{\boldsymbol{y}}^{(t)} = \boldsymbol{y}^{(t)} - \mathrm{E}\left[\boldsymbol{y}^{(t)}\right] - \sum_{i=1}^{t-1}\mathrm{Cov}\left(\boldsymbol{y}^{(t)}, \widetilde{\boldsymbol{y}}^{(i)}\right)\mathrm{Cov}\left(\widetilde{\boldsymbol{y}}^{(i)}\right)^{-1}\widetilde{\boldsymbol{y}}^{(i)} \tag{2.26}$$

where we define $\widetilde{\boldsymbol{y}}^{(i)} = \boldsymbol{y}^{(i)} - \mathrm{E}\left[\boldsymbol{y}^{(i)}\right]$. Equation 2.26 defines LIS. Substituting Equation 2.24 into Equation 23 yields the final form:

$$\boldsymbol{s}^{(T)} = \widehat{\mathrm{E}}\left[\boldsymbol{z}\middle|\boldsymbol{Y}^{(T)}\right] = \widehat{\mathrm{E}}\left[\boldsymbol{z}\middle|\widetilde{\boldsymbol{Y}}^{(T)}\right] = \mathrm{E}[\boldsymbol{z}] + \sum_{t=1}^{T}\widehat{\mathrm{E}}\left[\boldsymbol{z} - \mathrm{E}[\boldsymbol{z}]\middle|\widetilde{\boldsymbol{y}}^{(t)}\right] \tag{2.27}$$

Note, for each new sequence $\boldsymbol{y}^{(t)}$ observed, updating the estimate only requires inverting $\mathrm{Cov}(\widetilde{\boldsymbol{y}}^{(t)})$ as all other inversions are computed and can be stored from past steps. Returning to the motivating example, instead of inverting a single $(T * D) \times (T * D)$ matrix on the T^{th} iteration, a $D \times D$ matrix is inverted instead. But, Equation 2.26 demands the covariance of $\widetilde{\boldsymbol{y}}^{(t)}$, where we require the inverse of $(\mathbf{K} + \sigma_d^2\mathbf{I})$ for the KBGP previously described. These are fortunately the same thing: in the GP derivation we imposed $\mathrm{Cov}(\boldsymbol{y}) = \mathbf{K}$ in Equation 2.5, where $\boldsymbol{y}$ is our observation, as it is here.

Addressing the final undefined term, it can be shown that $\mathrm{Cov}\left(\boldsymbol{y}^{(t)}, \widetilde{\boldsymbol{y}}^{(i)}\right) = 0$ in Equation 2.26. This relies on the assumption that each collection is independent such that: $\mathrm{Cov}\left(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}\right) = 0$ for $i \neq j$. This result simplifies the incorporation of LIS, with the final results being:

$$\boldsymbol{s}^{(T)} = \widehat{\mathrm{E}}\left[\boldsymbol{z}\middle|\boldsymbol{Y}^{(\boldsymbol{T})}\right] = \mathrm{E}[\boldsymbol{z}] + \sum_{t=1}^{T} \mathrm{Cov}\left(\boldsymbol{z}, \widetilde{\boldsymbol{y}}^{(t)}\right)\mathrm{Cov}\left(\widetilde{\boldsymbol{y}}^{(t)}\right)^{-1}\widetilde{\boldsymbol{y}}^{(t)} \tag{2.28}$$

And in the GP notation established previously:

$$\boldsymbol{s}^{(t)} = \boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{Y}} = \boldsymbol{\mu}_{\mathbf{z}} + \sum_{i=1}^{t} \mathbf{k}^{\mathrm{T}}\,(\mathbf{K} + \sigma_d^2\mathbf{I})^{-1}\widetilde{\boldsymbol{y}}^{(i)} \tag{2.29}$$

where $\boldsymbol{y}^{(t)}$ is the t^{th} estimated source spectrum. Finally, the undefined $\boldsymbol{\mu}_{\mathrm{z}}$ can be estimated from the previous collection step. For example, the first collection assumes $\boldsymbol{\mu}_{\boldsymbol{z}} = 0$, as GPs do. The second collection will assert $\boldsymbol{\mu}_{\boldsymbol{z}} = \boldsymbol{s}^{(1)}$ and so on. Thus, the mean of the source distribution is updated after each collection step.

CHAPTER 3: ALGORITHM

3.1 ALGORITHM OVERVIEW

The algorithm for anomaly detection is broken into two parts: the first part is a GP for background removal on a spectrum to spectrum basis (GP-BR) utilizing a GP regression, and the second part uses LIS and the estimates from the first part to update a serial source spectrum estimate (LIS-SE). The GP-BR views only the most recent collection and reduces noise in the spectrum. The LIS-SE considers its previous estimate and the most recent GP-BR output. The estimated source distribution is then analyzed for anomalies. The following sections provide an overview of these distinct steps.

3.2 SINGLE SPECTRUM SOURCE AND BACKGROUND SEPARATION

The GP-BR is a GP used to reduce the presence of background in collected spectra leaving a greater source to background ratio for further analysis. Using the GP framework established previously, the input to this GP, x, is channel number and the output, t, is the number of counts attributed to background in channel x. A similar methodology for high count spectra is presented in detail in [12], and so an overview with highlighted differences is presented here. The GP-BR is broken into three stages: 1) defining training set and testing sets from the most recent collected spectrum, 2) GP training and prediction, and 3) spectrum reconstruction. 3.1 displays an example spectrum which would serves as an input to this stage. Figure 3.2 shows the ground truth breakdown between source and background counts. The details on spectrum generation are found in Section 4.1.

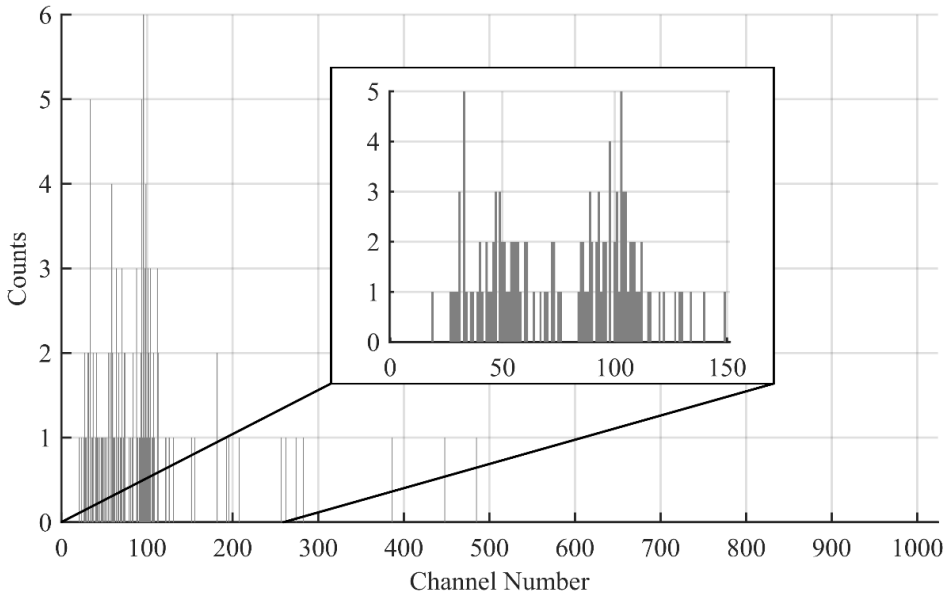

Figure 3.1: Example of a sparse gamma ray spectrum with a weak source peak with centroid at channel 100.

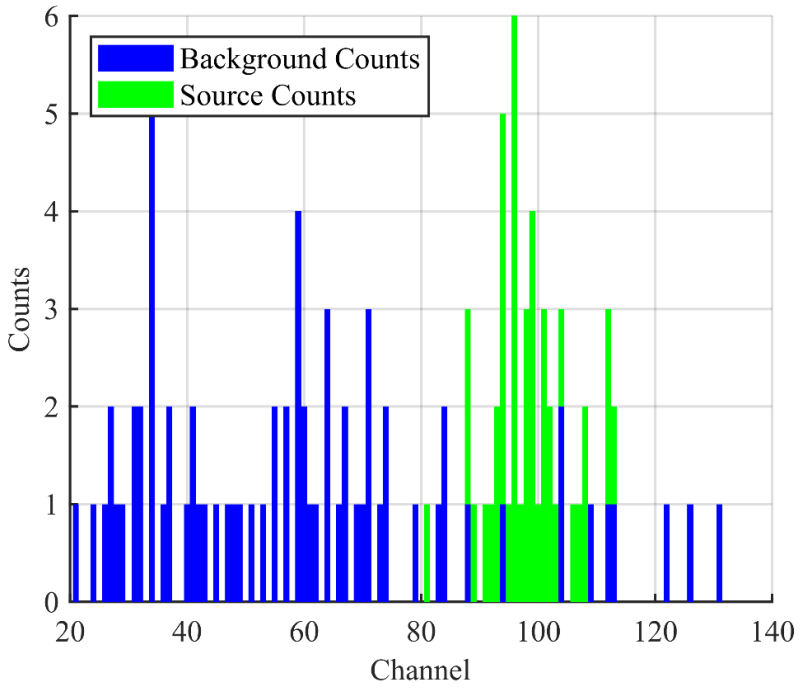

Figure 3.2: Input spectrum from Figure 3.1 with explicitly labelled source counts. The algorithm does not see this labelled information.

3.2.1 Training and testing sets

Stage one, called "spectral decomposition" in [12], involves decomposing a single collection into a training set representing channels containing predominately background counts and a testing set containing channels which have a mix of background and source counts. The training set, comprised of $\{\text{channel number}, \text{counts}\}$ pairs, must adequately represent background and is thus defined by the spectrum minima. All counts in the minima channels (including channels with zero counts) are assumed to be purely background counts [12, 27, 28]. The remaining $\{\text{channel number}, \text{counts}\}$ pairs form the test set and each contain a mix of background and source counts. The goal is to estimate how many counts belong to background in the test set. For extremely sparse data, however, this definition is insufficient since the majority of channels are zero-count (minima) and many of the non-zero count channels are adjacent to these. This leads to the common situation where all channels with non-zero counts are labelled as a nonminimum, leaving background to be zero in every channel.

To resolve this issue, two spectra are collected (1 second of data each for the application presented in this thesis) rather than one. The minimum check is performed on each 1 second of data independently. If a given channel d is labelled as a minimum in either of the spectra, then the associated channel in their summed 2 s spectrum is labeled as a minimum as well (only background). A channel with no source presence should not consistently contain counts on such a short time scale with a low sensitivity handheld detector, so using this technique to select against low count channels aids in defining a reliable training set. This process leaves us with a background training set of channels and a testing set of channels.

3.2.2 GP estimation

Now with testing and training sets, estimation commences. As mentioned previously, we omit training the hyperparameter training phase common for GP regression opting for a direct definition in order to reduce the time of estimation.

While the Matérn and Gaussian kernels tested in [12] were found to perform satisfactorily for their data, the samples considered here contain one tenth the counts. The data is extremely non-smooth and thus the semi-smooth Gaussian and Matérn kernels underperform. The locally-periodic kernel in Equation 2.21 is instead used as it better captures the discontinuous nature of the data over short length scales. The hyperparameters ℓ and p are fixed at implementation. The parameter ℓ is set to the full width at half maximum (FWHM) value at the peak resolution of the detector. The justification of this selection is that the detector cannot resolve below this threshold, and so the tails of any photopeak will just fall within this characteristic length. The periodicity p is set equal to 2 such that the discrete changes in counts between neighboring channels can be accounted for. Setting $p = 2$ (as seen in Figure 2.1c) increases the covariance between every second channel while selecting against immediate neighbors. The hyperparameter σ_d^2 is determined by computing the maximum variance in counts achieved over the characteristic length.

With a tuned kernel, the training set, and testing set, Equation 2.12 yields the estimated mean number of background counts in mixed channels while Equation 2.13 provides the variance for each of those estimations.

3.2.3 Source and background decomposition

The final stage decomposes the full spectrum into estimated background and estimated source spectra using the estimated mean background counts in mixed channels. Each tested channel

undergoes the variance check and discretization from [12]. The variance check uses the variance computed from Equation 2.13 to test whether the recorded counts fall within two standard deviations of the mean background estimate; if so, then all counts in that channel are assigned to background, else only the estimated amount are and the remainder are assigned to source. Discretization, the rounding of output to whole numbers, is necessary as the output of the GP is not discrete but counts are.

Since the spectra are so sparse, however, an additional step is required to help offset the problem of many nearby minima (zero count channels in particular) causing the GP to place channels with few counts into the background spectrum despite having a high-count neighbor channel. For any test channel which had all counts placed into the background estimate after the variance check, if its immediate neighbors still contain source counts, then only the estimated counts are set as background instead of all. Thus, the output consists of an estimated background spectrum and estimated source spectrum each of length D. This output can be seen in Figure 3.3.

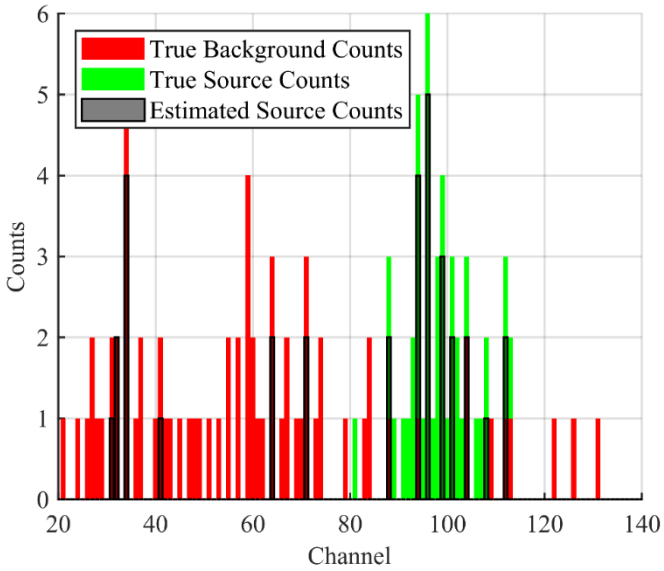

Figure 3.3: Example spectrum with estimated source counts from the GP-BR labelled in blue. These estimated counts are the input into sure LIS-SE.

3.3 TIME-SERIES UPDATES OF SPECTRA ESTIMATES

Relying on the background removed output from the GP-BR, the LIS-SE is for the task of source distribution estimation and anomaly detection.

Each background removed collection $\boldsymbol{y}^{(t)}$can be viewed as a sampling from the true source distribution $\boldsymbol{z}$, and the LIS-SE uses these samples to build a mean estimate for it. Each new background removed collection $\boldsymbol{y}^{(T+1)}$ is added to a sequence of observations $\boldsymbol{Y}^{(\boldsymbol{T+1})} = \{\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(T)}, \boldsymbol{y}^{(T+1)}\}$ and all channels $\boldsymbol{x}$ need be tested for source presence. Thus, the estimate for the true source distribution $\boldsymbol{z}$ is computed with Equation 2.28. Utilizing the LIS scheme for updates reduces the complexity of the problem as compared to an ever-growing observations vector. The estimation cycle is separated again into three stages: 1) prepare the training set with weighting, 2) prediction with LIS, and 3) source estimation and anomaly detection.

3.3.1 Training set preparation

In order to build a training set, stage one processes the background removed spectar from the GP-BR by weighting each channel with the inverse of the number of neighboring zeros. Since higher count density is expected in photopeak regions, this process helps select against poorly filtered background peaks. This weighting scheme counts the number of zeros around channel d with a window of size ℓ across all T collections; let this vector be noted as $\boldsymbol{w}$ where the d^{th} element is:

$$w_d = \sum_{t=1}^{T} \sum_{i=d-\frac{\ell}{2}}^{d+\frac{\ell}{2}} \delta\left(\boldsymbol{y}^{(t)}(C_i)\right) \tag{3.1}$$

The counts in each channel are then divided by the square of this zero-weight and normalized across each collection:

$$\boldsymbol{y}_d^{(t)\prime} = \frac{\boldsymbol{y}_d^{(t)}}{\boldsymbol{w}_d^2}\left(\sum_{i=1}^{D} \frac{\boldsymbol{y}_i^{(t)}}{\boldsymbol{w}_i^2}\right)^{-1} \tag{3.2}$$

where D is the number of channels. Following the LIS requirement, our training set is then:

$$\tilde{\boldsymbol{y}}^{(t)} = \boldsymbol{y}^{(t)\prime} - \mathrm{E}\left[\boldsymbol{y}^{(t)\prime}\right] \tag{3.3}$$

where $\mathrm{E}\left[\boldsymbol{y}^{(t)\prime}\right]$ is directly computable as the average counts in each channel. This weight vector is updated after every collection.

3.3.2 Estimation with LIS

Stage two uses Equation 2.28 to compute the mean of the estimated source distribution. The Gaussian kernel function Equation 2.16 with characteristic length ℓ set equal to the detector resolution was chosen for the source estimation regression task because the expected source distribution will be smooth. The summation in Equation 2.28 is over T collections and where $\boldsymbol{\mu_z} = \boldsymbol{y}^{(T-1)}$ for $t > 1$. The hyperparameter σ_d^2 is also defined at this point as:

$$\sigma_d^2 = \sigma_h^2 + \sigma_{spectrum}^2 \tag{3.4}$$

where σ_h^2 is the variance in counts C in channel d across all collections T:

$$\sigma_h^2 = \frac{1}{T}\sum_{t=1}^{T}\left(\boldsymbol{y}_d^{(t)}(C_d) - \mathrm{E}\left[\boldsymbol{y}_d^{(t)}(C_d)\right]\right)^2 \tag{3.5}$$

And $\sigma_{spectrum}^2$ is the variance across all channel measurements:

$$\sigma_{spectrum}^2 = \frac{1}{TD}\sum_{t=1}^{T}\sum_{d=1}^{D}\left(\boldsymbol{y}_d^{(t)}(C_d) - \mathrm{E}[\boldsymbol{y}(C)]\right)^2 \tag{3.6}$$

3.3.3 Source estimation

The final stage takes the mean estimate after each new collection to build an updated source distribution estimate. Since the output of the LIS-SE, $\boldsymbol{s}^{(T)}$, can be negative, but counts cannot be, any channel containing negative values is set equal to zero. Negative estimates occur in channels predicting approximately zero but which are near channels with higher count rates. Then, since the

training data was scaled down with weighting, the output spectrum $\boldsymbol{s}^{(T)}$ is scaled by the cumulative counts in the background removed collections, $\boldsymbol{s}_{BR}^{(t)}$:

$$\hat{\boldsymbol{s}}^{(T)} = \left(\frac{\boldsymbol{s}^{(T)}}{\sum_{d=1}^{D} s_d^{(T)}}\right) \sum_{d=1}^{D} \sum_{t=1}^{T} \boldsymbol{s}_{NR}{}_d^{(t)} \tag{3.7}$$

Here, $\hat{\boldsymbol{s}}^{(T)}$ updates after each collection and provides a smooth estimate for the source distribution over all detector channels.

3.4 ANOMALY DETECTION THRESHOLDING

The features of $\hat{\boldsymbol{s}}$ are used to determine location of anomalies by acquiring the height (counts) and location (channel) of maximums. Each maximum is then tested against a Gaussian peak as the LIS-SE produces Gaussian features due to choice of kernel function. An area $[d_{lower}, d_{upper}]$ is drawn around each maximum channel where: d_{lower} is the first point left of the maximum where the value is equal to or less than one percent of the peak's height, reaches zero, or begins to rise again; and d_{upper} is the first point right of the maximum which meets the same criteria. A Gaussian distribution is then fit to the cumulative output of the GP-BR source estimate over this range:

$$\boldsymbol{s}_{NR}^{(1:T)} = \sum_{t=1}^{T} \boldsymbol{s}_{NR}{}_{[d_{lower}, d_{upper}]}^{(t)} \tag{3.8}$$

If the R^2 value computed for this fit is less than a predefined threshold, then the peak is ignored for this collection. This is referred to as the R^2 test. Poor fits are caused by either a small number of counts or peaks formed from background counts which leaked through from the GP-BR. Peaks formed from leaked background are not Gaussian and so fit poorly to a Gaussian distribution. If the R^2 value is greater than a determined threshold, then this peak is considered and subjected to a density test. If two-thirds of the area under the curve of $\hat{\boldsymbol{s}}$ is within

$[d_{lower}, d_{upper}]$, then the peak is considered an anomaly. The relative area under the curve over a specific window is referred to as “density”.

CHAPTER 4: EXPERIMENTS

4.1 DESCRIPTION OF THE DATA

The background test data was collected with a Kromek D3S gamma-ray detector on the University of Illinois at Urbana-Champaign campus with no sources other than NORM present. This background set consists of 14077 sample spectra, each 1 s collections, with an average count rate of 40 ± 16 counts/s. An artificial source spectrum was synthesized via a statistical model. With a given source activity a and dwell time t, for each spectra, a Poisson random number was computed representing the total emissions λ as $\lambda \sim Pois(a * t)$. With a defined distance d meters from the source and the face dimensions of the detector (0.5 in $\times$ 2.54 in), the geometric efficiency ϵ_g was calculated and used to compute the number of counts captured by the detector as $c = \epsilon_g \lambda$. The standard deviation of the detected count spread σ was computed using the FWHM of the detector. Detected counts c were then placed into the source spectrum around channel d according to a random Normal distribution following $N(d, \sigma^2)$. This peak is computed for each second of dwell time independently. This is a simplistic mechanism for injecting photopeaks into background spectrum and neglects detector physics in the binning of photons, such as internal Compton scattering. Such a simplification is adequate here as we are only concerned with photopeak versus non-photopeak counts in defining anomalies with this methodology. Taken with the fact that the Compton continuum has a large energy range with semi-random count placement, it would be interpreted as background with this method.

Since these tests are meant to be performed under sparse conditions, a source count rate of 20 counts/s, half that of the background count rate, was selected. This roughly equates to an activity of 2.2 mCi at a distance of 10 m. The source peak was placed at channel 100, just

overlapping the noise region. For source peaks at higher channel numbers, the source counts are more easily distinguished from background. For source peaks at lower channel numbers, the background counts add to source peaks consistently, reducing the number of local minima, reinforcing the source peak location. As such, this placement represents the worst-case scenario.

The source spectrum is then injected into the background spectrum such that a ground truth about background and source spectrum are known for each collection.

4.2 DESCRIPTION OF EVALUATION METHODS

4.2.1 Evaluation of GP-BR

The validation of the background removal algorithm consisted of tests with and without source present as described in section 4.1. The metric of each test was how well the background and source spectra were separated as quantified by the correlation coefficient between the respective true and estimated spectra. Each test consisted of thirty independent trials using 2 s collections each. These results are compared against the results presented in [12] since these are functionally similar algorithms with ours being modified for extremely sparse spectra. The algorithm from [12] was employed in full except for the hyperparameter tuning, opting instead for the fixed parameter approach to reduce the variance in performance since this method is not well suited for sparse data.

4.2.2 Evaluation of LIS-SE and Anomaly Detection

The LIS-SE tests consisted of, like in the previous section, one experiment with a source and another without. Each of these experiments consisted of 100 trials each taking 15 successive collections of 2 s measurements to make use of the updating scheme available to the LIS-SE. Each collection was first passed through the GP-BR to reduce the noise. The results of the LIS-SE can

be reviewed qualitatively by observing the resulting source spectrum and comparing against the ground truth source spectrum. In order to use anomaly detection, a threshold for the R^2 test first needs to be decided. This is selected based on ROC curve analysis such that true positives are maximized while false negatives are minimized. With the anomaly detection thresh decided, the true positive anomaly identification rate is constructed to describe the accuracy of predictions. Finally, the true and estimate source counts is compared.

4.3 IMPLEMENTATION IN SOURCE LOCALIZATION TASK

Utilization of this methodology is encouraged in source localization schemes where source and background counts are needed from an algorithmic standpoint. As such, this framework has been incorporated into a reinforcement learning algorithm for navigating a single detector system presented in [29]. For testing, a simulated environment of a $10\,m \times 10\,m$ area with grid points every $1\,m$ was initialized with origin $[0,0]$ in the bottom left corner. The detector agent was initialized to position $[1,10]$ (upper left corner); the source was initialized to $[8.5, 2.5]$ (bottom right corner). Test source strength was set to 2000 cnts/s at a 1 m distance and an appropriate source count rate was initialized to each grid point. A random Poisson number is generate whenever the detector enters a specified grid point with mean equal to that grid point's initialized count rate. This yielded ~35 source counts detected at the initialization position after a 2 s collection. Background and source data were constructed based on section 4.1 noting that 2 background spectra are used to accommodate a 2 s dwell period. A wall of random length is placed extruding from a random location around the perimeter. This wall blocks source counts and cannot be moved through.

Here, we compare the localization speed while using gross counts, as in the original paper, against using anomalous counts as defined by the anomaly detection algorithm presented in this thesis. Two cases were considered for the gross count trials: an instance where the background count rate was 0, and an instance when the background count rate was an average of 40 cnts/s (about that present in the background spectrum described in section 4.1). A usage note of this algorithm is that the anomaly threshold needs to be passed for source counts to be yielded. As such, navigation uses gross counts until an anomaly is detected in which case counts from the source peak region are used. 300 trials were conducted for each of gross count zero background case, gross count some background case, and anomalous counts case.

CHAPTER 5: RESULTS AND DISCUSSION

5.1 ESTIMATION RESULTS VIA GP-BR

For pure background, the proposed GP-BR method provides an average correlation coefficient of 0.76 ± 0.08 between the estimated background and true background spectrum over the thirty trials; the background estimation scheme from [12] yields a correlation coefficient of 0.36 ± 0.14 between estimated background and true background. The correlation coefficient reported in [12] for the same set of tests was 0.816, the key difference in performance being the sparsity of the data used. Without implementing the additional strategies for selecting against background, the sheer abundance of zero count channels leads channels with any counts being improperly placed in the source spectrum.

With a 2.2 mCi source placed 10 m from the detector, thirty 2 s source spectra were created. These were then injected into randomly selected background spectra. The average correlation coefficient of estimated background to true background spectrum was 0.629 ± 0.15; and the estimated source to true source spectrum average correlation coefficient was 0.67 ± 12. Implementing the GP from [12] as before yields a background correlation coefficient of 0.33 ± 13 and a source correlation coefficient of 0.60 ± 0.08.

These tests indicate that the proposed method for background and source spectra separation improves over [12] in the case of extremely sparse data and fixed hyperparameters. However, there is still sub-optimal estimation of source spectrum as indicated by the relatively low correlation coefficients. This motivates the need for the LIS-SE.

5.2 ANOMALY DETECTION RESULTS VIA LIS-SE

First, we can observe the performance of the LIS-SE module by inspecting the output over the course of the trials. The evolving output can be seen in Figure 5.1 over the course of 10 collections. In observing the output from the LIS-SE in Figures 5.2, 5.3, and 5.4 with the context of the labelled spectrum data, two notable peaks form: a peak centered near channel 100 correlating to the source peak and another around channel 40 correlating to a background peak. Figures 5.2, 5.3, and 5.4 correspond to collection numbers 1, 5, and 15, respectively. The background peak is present with or without source presence. While some smaller peaks are identifiable, the results presented here focus on the noise peak which demonstrated the strongest chance to incorrectly pass the anomaly detection criteria, e.g. the one with the largest R^2.

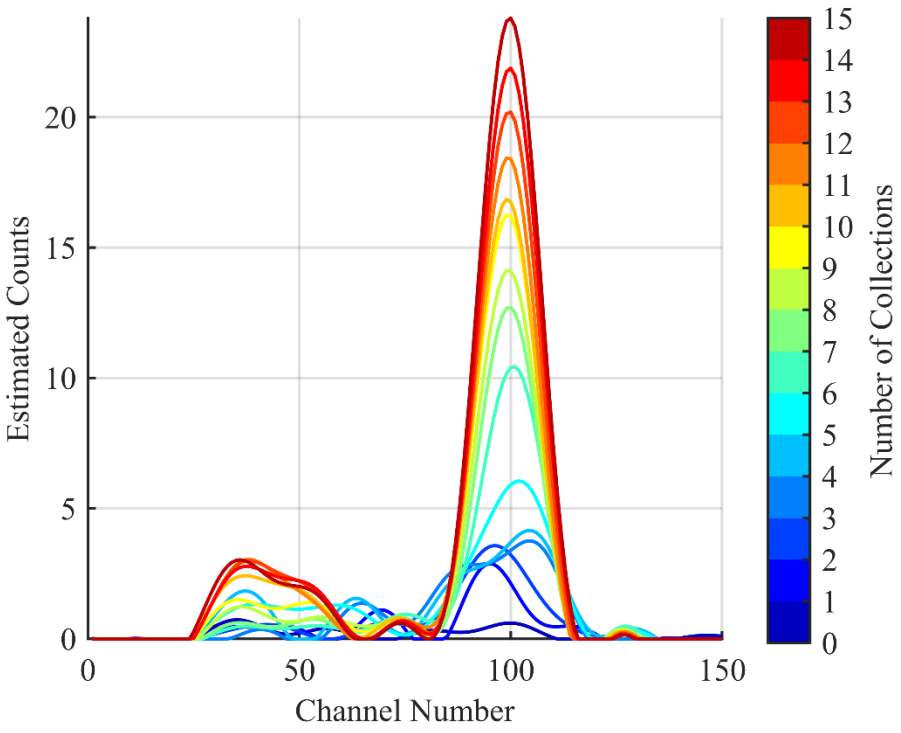

Figure 5.1: Series of the estimated source distribution after each of 10 collections. Note the source peak (channel 100) raises at a much faster rate than the noise peaks (all other peaks).

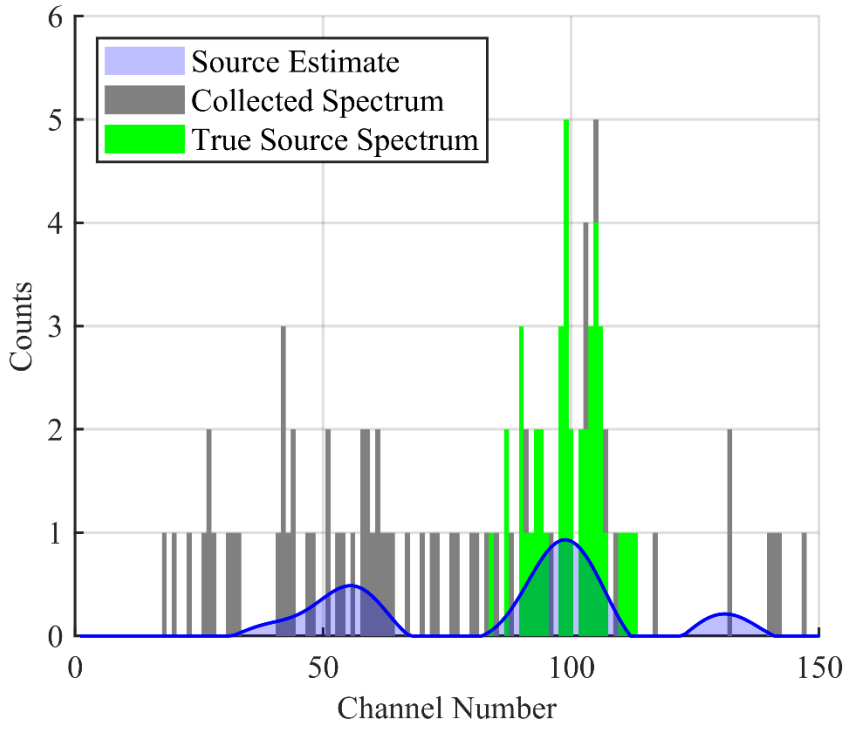

Figure 5.2: Estimated source spectrum after 2 s of data. The estimated source distribution is given in blue, the true background spectrum is given in grey, and the true source spectrum is given in green.

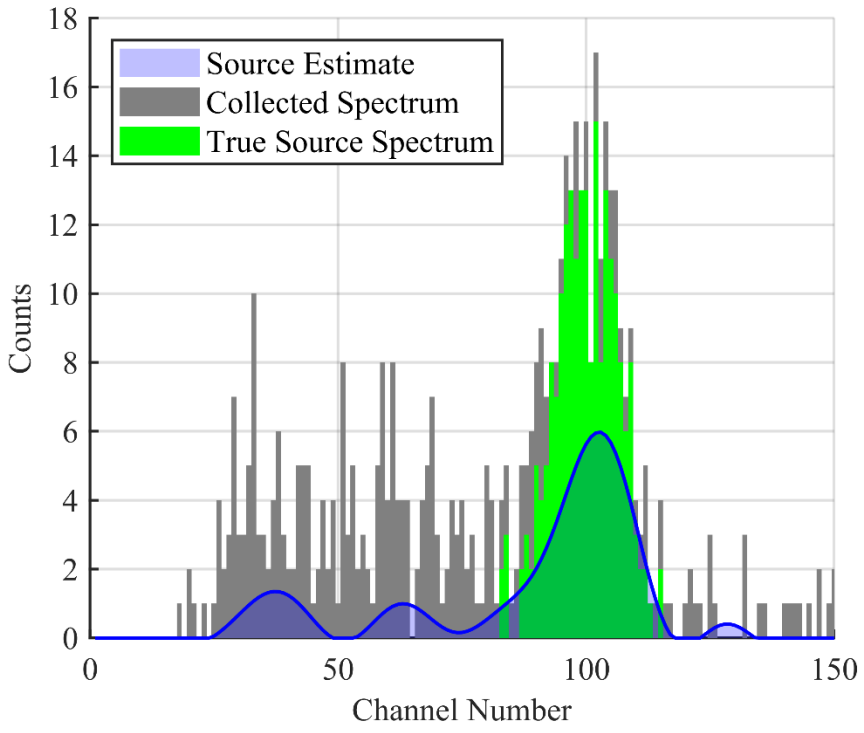

Figure 5.3: Estimated source spectrum after 10 s of data. The estimated source distribution is given in blue, the cumulative collected spectrum is given in black, and the true source spectrum is given in green.

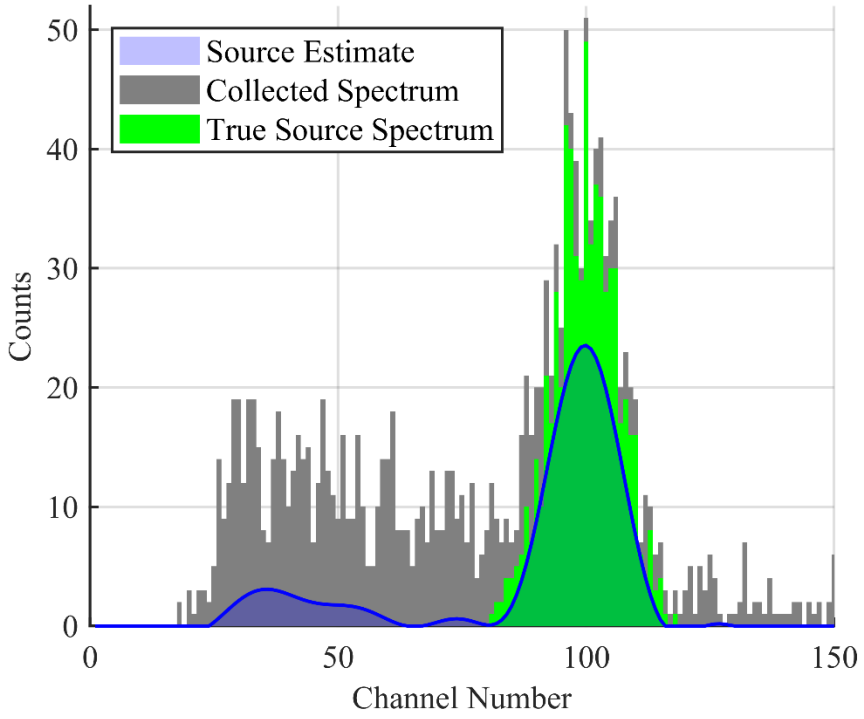

Figure 5.4: Estimated source spectrum after 30 s of data. The estimated source distribution is given in blue, the cumulative collected spectrum is given in black, and the true source spectrum is given in green. Note the estimated noise peak as compared to the actual counts in corresponding channels.

Both the noise and source peaks form Gaussian-like peaks due to the selection of kernel. The R^2 coefficient used for the anomaly detection test rose with each successive trial for the source peak as the number of counts become sufficient enough to be represented as a Gaussian as seen in Figure 5.5. Here the most prominent background R^2 value only surpassed the photopeak R^2 value in early collections. Presented are the average values over all 100 trials. It also demonstrates that the noise leaked through GP-BR becomes less Gaussian over time as the value falls over the

successive collected spectra. Relative density accumulated within the true peak region consistently while the noise region decreased in estimated source presence as seen in Figure 5.6.

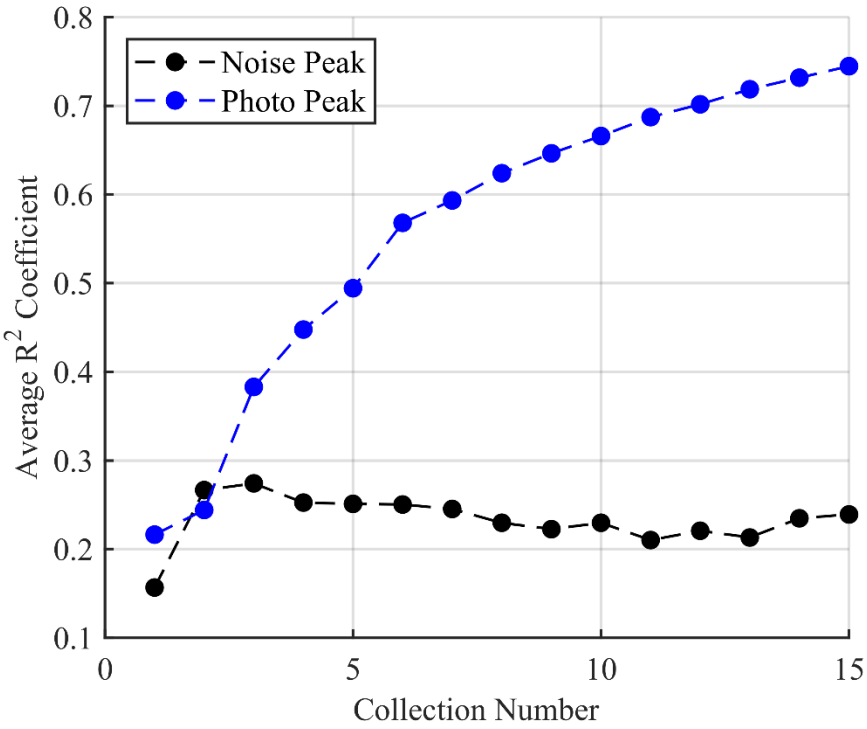

Figure 5.5: The average R^2 test results for the 100 trials with source present. The source peak is increasingly Gaussian with each successive collection whereas the most convincing noise peak is poorly fitted over all collections.

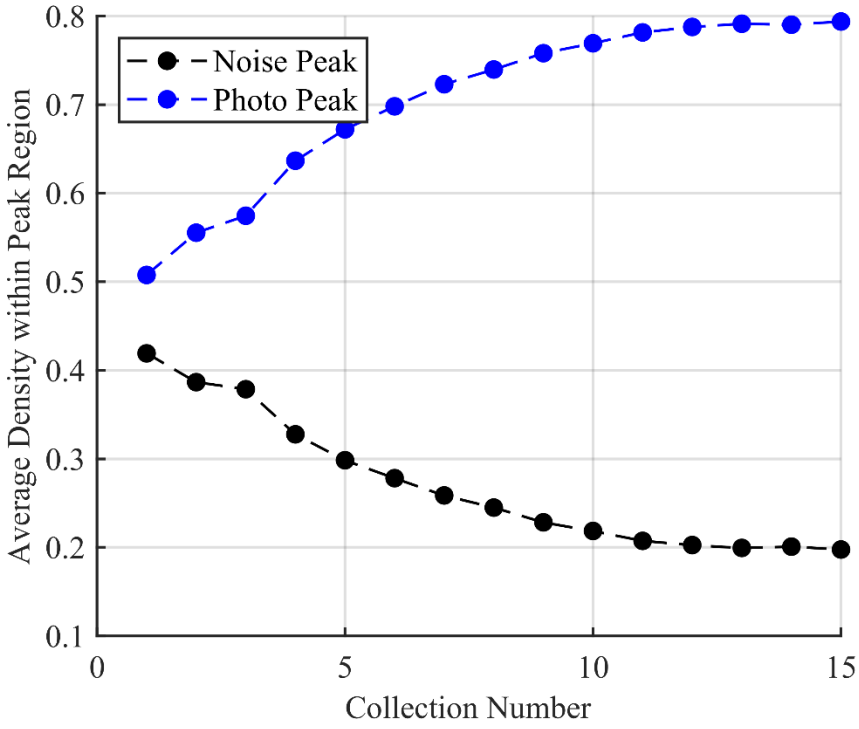

Figure 5.6: The average density of the noise and source peaks over the 100 trials with source present. The density within the source region increases with each successive collection while that of noise regions fall.

For the trials with no source, every trial passed the density test. This makes sense as only a single peak from leaked background is present and so would necessarily contain the majority of the density. As such, the R^2 test did a good job selecting again noise peaks as their distribution was not well captured by a Gaussian fit. The R^2 value of the best scoring noise peak can be seen in Figure 5.7 and the density of the largest noise peak in Figure 5.8.

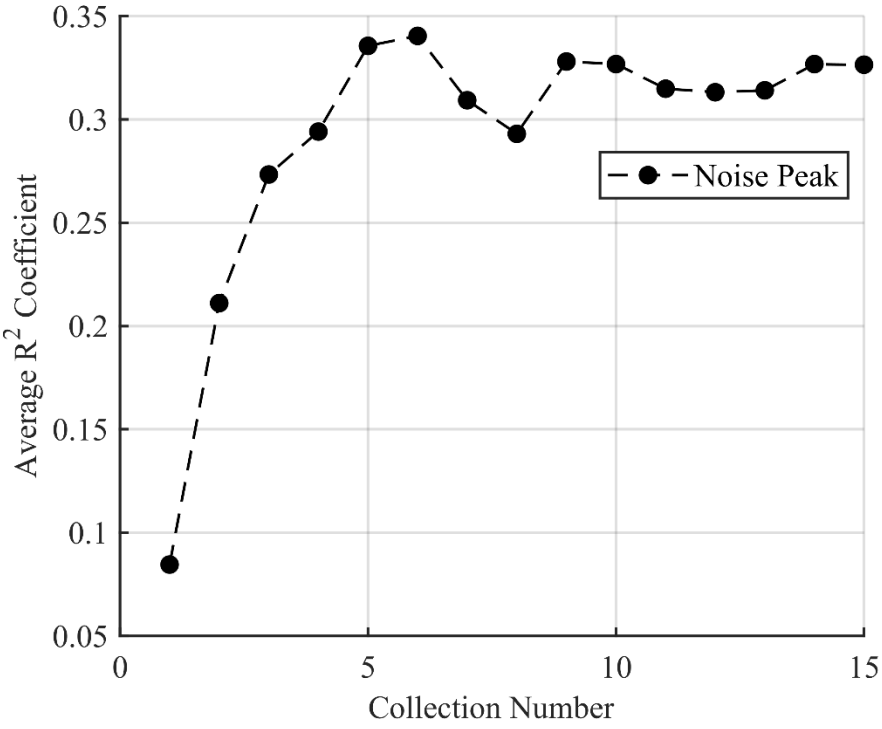

Figure 5.7: The average R^2 test results for the 100 trials with no source present. The average R^2 value for the best performing noise peak mirrors that of Figure 5.5, illustrating that the noise peak shape is independent of source presence.

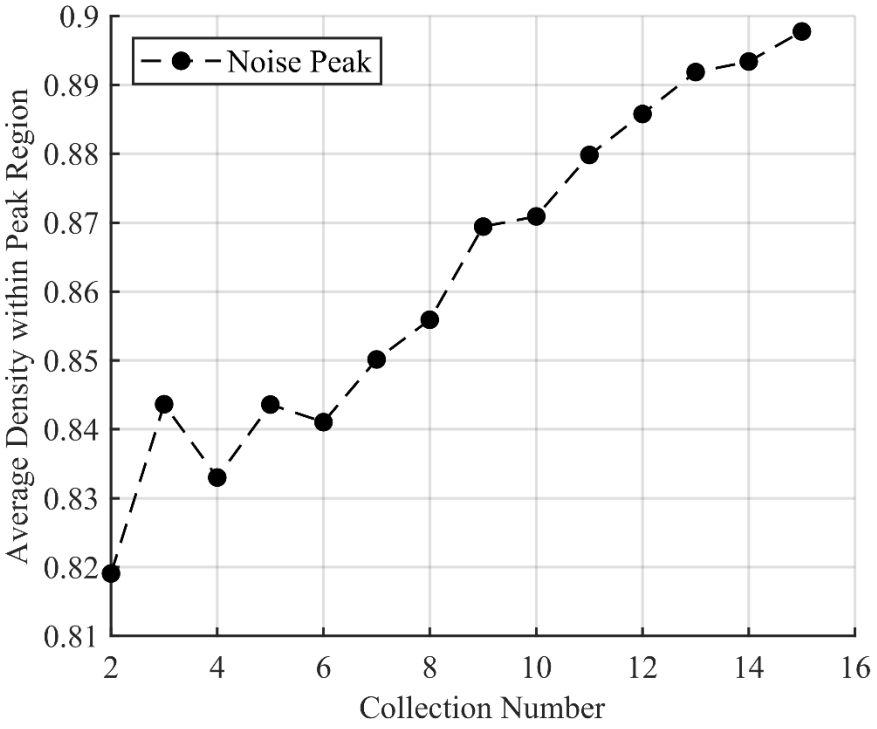

Figure 5.8: The average density of the noise peak over the 100 trials with no source present. Noise density is accumulating, which is undesirable but is expected since no source is present.

The true positive rate for anomaly detection across the full algorithm – using the GP-BR for noise reduction and the LIS-SE for source estimation – with an R^2 threshold of 0.65 and averaged over all trials was 0.93. This can be improved upon by noting that these tests were conducted with a stationary detector. If paired with a surveying algorithm, as intended, the amount of information available to the algorithm will increase with time and should improve this accuracy. The ROC curve for anomaly detection while adjusting the R^2 threshold can be seen in Figure 5.9. The area under the ROC curve was 0.902, demonstrating a good estimation process.

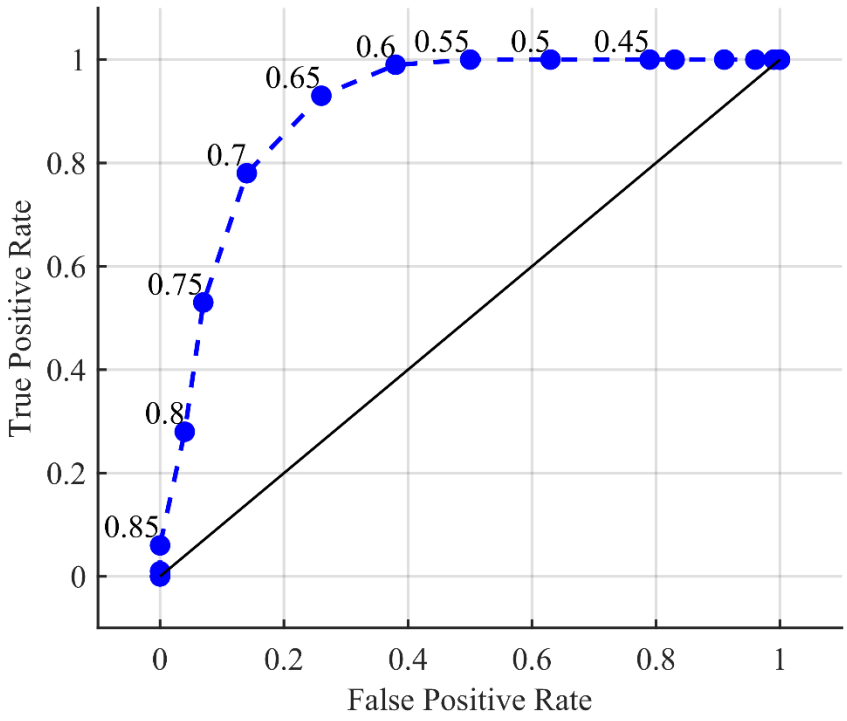

Figure 5.9: ROC curve for anomaly detection. False positive defined as claiming an anomaly is present when none are. True positive defined as claiming an anomaly is present when one is. The labels refer to the threshold set on the R^2 test. If the R^2 value is above this threshold, then the peak is considered anomalous.

5.3 BACKGROUND REMOVAL IN SOURCE LOCALIZATION TASK

Over the 300 trials of the three cases, the best performance occurred when the navigation was coupled with anomaly detection. The results can be seen in Figure 5.10 where the closer the curve is to the upper-left corner, the better. This plot represents the percentage of the 300 runs for which the source was localized in a certain number of steps. We see that after about 350 steps, all trials for each case had found the source. The largest difference was in the 10 to 200 step range where about 90% of trials had found the source after 40 steps in the 40 background cnts/s case while the other two cases were above 95%. When using gross counts with some background present, the

agents took an average of 28.4 steps to localize the source; using gross counts with no background present, the agent took an average of 23.1 steps to localize the source; using the coupled anomaly detection algorithm and anomalous counts, the agent took an average of 20.7 steps. It is not intuitive that the coupled performance outperforms a case were no background is present. The explanation is that at far distances from the source, such as the initialization positions, occasionally no or very few source counts are detected, leading the RL navigation to make steps in random directions. This is exasperated by the fact that the RL algorithm was trained with background, leading to some unencountered states when background is absent.

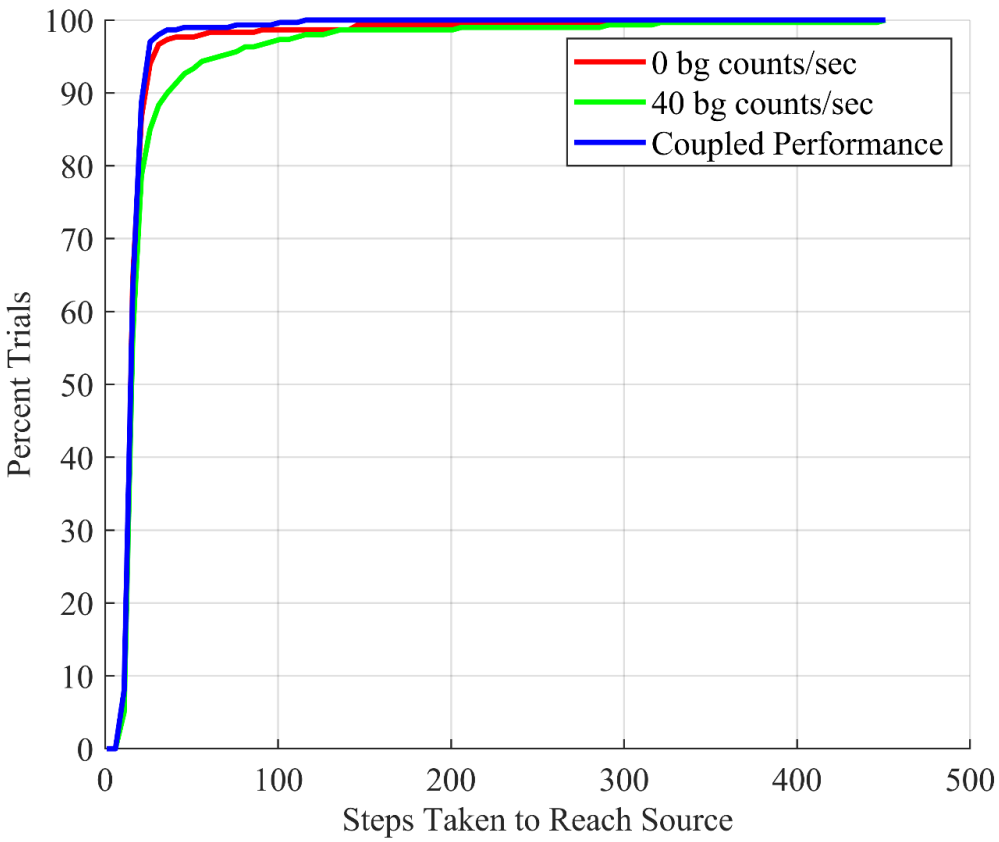

Figure 5.10: Comparison of localization speed for each background and count consideration case.

A depiction of the anomaly detection algorithm process for a single trial where no wall is present is displayed in Figure 5.11 and an average over 30 trials where no wall is present can be seen in Figure 5.12. In the single trial case, an anomaly is detected after the 7th collection period, after which a very close match of background and source counts can be observed. This leads to the navigational algorithm making movement decisions based on high quality information. In the averaged trials, we see than prediction becomes accurate and stays accurate after 8.5 steps. Further, after an anomaly is detected, estimated gross source counts and true gross source counts have an average correlation of 0.998, whereas estimated gross background counts and true gross background counts have an average correlation of 0.876. This is very strong agreement.

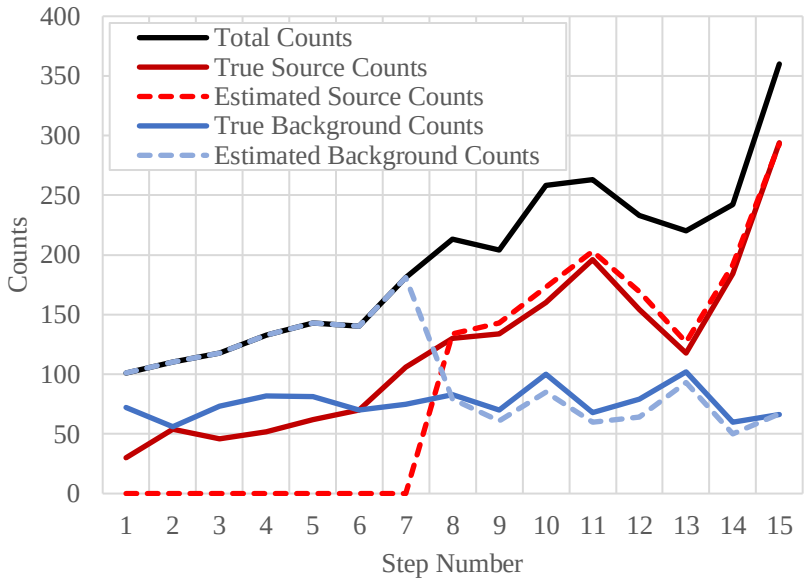

Figure 5.11: Count estimations during a single navigation at each step. Note that the anomaly detection turns on after step 7, resulting in high quality source and background count predictions.

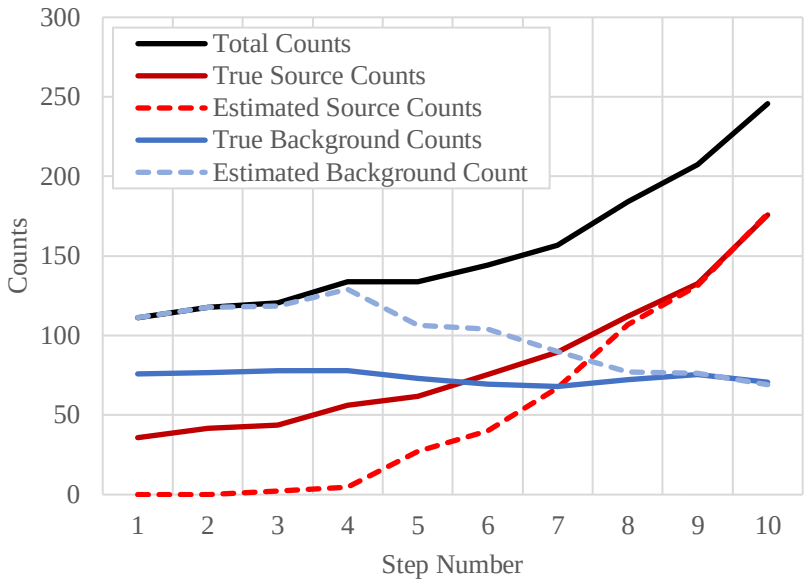

Figure 5.12: Averaged count estimations over 30 localization trials. All localization trials had anomalous counts detected at step 9.

The demonstrations presented in this section display the usefulness and accuracy of the proposed methodology for single detector, source localization algorithms. Here, localization times were shown to improve over gross counts usage.

CHAPTER 6: CONCLUSION AND FUTURE WORK

Through a two-stage statistical model, we have developed an algorithm for attaining a source spectrum estimate and detecting anomalous sources in sparse gamma-ray spectra scenarios. Using a GP for background removal provides a consistent, albeit unoptimized, source and background spectrum estimates. Here, the background count placement agreed with the ground truth to yield a correlation coefficient of 0.76 in pure background and 0.63 when a weak source was present. From successive sampling, however, as one would encounter in source surveying tasks, we are able to construct a good estimate of the source spectrum via a LIS scheme, accurately identify source peaks with a 0.903 true positive rate, and provide exceptional source and background count rate reconstruction. Further, we have demonstrated its application in a source surveying scenario, where the anomalous count estimate improved the localization speed of a gamma-ray source and improved the quality of information used by the navigational protocol.

While we demonstrated that this procedure is successful for a 2.2 mCi source 10 m from a low sensitivity detector whose photopeak just overlaps a noise dominated region, this algorithm can be applied to any number of combinations of source type, detector type, and distance. Most notably, this algorithm requires no training prior to use and is independent of gamma detector specifics except for a single hyperparameter governed by resolution. As such, it can be immediately incorporated into any sparse data gamma analysis for which a source location (peak, region, etc.) or count estimates are required.

The demonstrated approach is successful in identifying a single, weak source peak, but future work should amend the anomaly thresholding technique used. Namely, the thresholding criteria for the R^2 test is essentially a learned parameter, something we are trying to avoid in this

approach. Additionally, the density test may be unnecessary as it ultimately restricts anomaly detection to a single, large photopeak. This being said, the LIS-SE and GP-BR presented are insensitive to spectra with multiple photopeaks, so such amendments are feasible without rework the bulk of the algorithm.

Since the LIS-SE algorithm can be interpreted as a Bayesian take on GP regression, it should be possible to provide a confidence/variance metric for every estimated point. In the current implementation, since every channel contains training and testing data, however, this results in equal variance everywhere except channel extremes where there are no neighbor present. A reimagining of this updating scheme could provide better source spectrum estimates with confidence metrics. Similarly, every spectrum collected and fed into the algorithmic pipeline is considered equally important. While this is the case for a stationary detector collecting serial samples, it would make sense that in source surveying settings, more recent collections should be weighted more heavily. Finally, the proposed methodology is highly tuned for sparse data. Identifying the point at which such an algorithm is no longer viable would prove immensely helpful in securing confidence in the efficacy of its outputs.

In this thesis, I have achieved the desired goals of developing a computationally efficient anomaly detection algorithm tuned for source surveying protocols. Most importantly, it is free of prior training and all hyperparameters are physically defined.

REFERENCES

[1] Ristic B, Gunatilaka A. Information driven localisation of a radiological point source. Information fusion. 2008 Apr 1;9(2):317-26.

[2] Liu Z, Abbaszadeh S. Double Q-Learning for Radiation Source Detection. Sensors. 2019;19:960.

[3] Jarman KD, Runkle RC, Anderson KK, Pfund DM. A comparison of simple algorithms for gamma-ray spectrometers in radioactive source search applications. Applied Radiation and Isotopes. 2008 Mar 1;66(3):362-71.

[4] Knoll, Glenn F. Radiation detection and measurement. John Wiley & Sons, 2010.

[5] Ely JH, Kouzes RT, Geelhood BD, Schweppe JE, Warner RA. Discrimination of naturally occurring radioactive material in plastic scintillator material. IEEE Transactions on Nuclear Science. 2004 Aug;51(4):1672-6.

[6] Kouzes RT, Ely JH, Geelhood BD, Hansen RR, Lepel EA, Schweppe JE, et al. Naturally occurring radioactive materials and medical isotopes at border crossings. In2003 IEEE Nuclear Science Symposium. Conference Record (IEEE Cat. No. 03CH37515) 2003 Oct 19 (Vol. 2, pp. 1448-1452). IEEE.

[7] Philips GW, Nagel DJ, Coffey T. A Primer on the detection of nuclear and radiological weapons. National Defense University Washington DC Center for Technology and National Security Policy; 2005 May.

[8] "Cesium-137." *Oncology Medical Physics*, oncologymedicalphysics.com/cesium-137/.

[9] Kirkpatrick JM, Young BM. Poisson statistical methods for the analysis of low-count gamma spectra. IEEE Transactions on Nuclear Science. 2009 Jun;56(3):1278-82.

[10] Fischer R, Hanson KM, Dose V, von Der Linden W. Background estimation in experimental spectra. Physical Review E. 2000 Feb 1;61(2):1152.

[11] Fagan DK, Robinson SM, Runkle RC. Statistical methods applied to gamma-ray spectroscopy algorithms in nuclear security missions. Applied Radiation and Isotopes. 2012 Oct 1;70(10):2428-39.

[12] Alamaniotis M, Mattingly J, Tsoukalas LH. Kernel-based machine learning for background estimation of NaI low-count gamma-ray spectra. IEEE Transactions on Nuclear Science. 2013 Jun;60(3):2209-21.

[13] D3S ID wearable RIID gamma neutron detector. Kromek Group. [cited 14 April 2019]. Available from: https://www.kromek.com/product/d3s_riid/.

[14] Liu Z, Abbaszadeh S, Sullivan CJ. Spatial-temporal modeling of background radiation using mobile sensor networks. PloS one. 2018 Oct 19;13(10):e0205092.

[15] Advanced Radioactive Threat Detection System Completes First Large-Scale Citywide Test [Internet]. Defense Advanced Research Projects Agency (DARPA). 2016 Nov 10 [cited 2019 Feb 15], Available from: www.darpa.mil/news-events/2016-10-11.

[16] Fischer A, Wrobel M. Sensing [Internet]. Defense Advanced Research Projects Agency (DARPA). 2017 June 15 [cited 2019 Feb 15]. Available from: www.darpa.mil/attachments/sensingfischerwrobel-a.pdf. Slide. 15

[17] Romanchek GR, Liu Z, Abbaszadeh S. Kernel-based Gaussian process for anomaly detection in sparse gamma-ray data. Plos one 15.1 (2020): e0228048.

[18] Bishop C, Pattern Recognition and Machine Learning. New York, NY: Springer-Verlag; 2016. p. 291-324.
[19] Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. Cambridge, MA: MIT Press; 2006.
[20] Ayat NE, Cheriet M, Suen CY. Automatic model selection for the optimization of SVM kernels. Pattern Recognition. 2005 Oct 1;38(10):1733-45.
[21] Duvenaud D. Automatic model construction with Gaussian processes [dissertation]. Cambridge, UK: University of Cambridge; 2014.
[22] Krishnamoorthy A, Menon D. Matrix inversion using Cholesky decomposition. In2013 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA) 2013 Sep 26 (pp. 70-72). IEEE.
[23] Seeger, Matthias. Gaussian processes for machine learning. International journal of neural systems 14.02 (2004): 69-106.
[24] Csató, Lehel, and Manfred Opper. Sparse on-line Gaussian processes. Neural computation 14.3 (2002): 641-668.
[25] Tipping, Michael E. "Sparse Bayesian learning and the relevance vector machine." Journal of machine learning research 1.Jun (2001): 211-244.
[26] Hajekm B. Random Processes for Engineers. Cambridge, UK: Cambridge University Press; 2015.
[27] Burgess DD, Tervo RJ. Background estimation for gamma-ray spectrometry. Nuclear Instruments and Methods in Physics Research. 1983 Sep 1;214(2-3):431-4.
[28] Tervo RJ, Kennett TJ, Prestwich WV. An automated background estimation procedure for gamma ray spectra. Nuclear Instruments and Methods in Physics Research. 1983 Oct 15;216(1-2):205-18.
[29] Romanchek GR, Liu Z, Abbaszadeh S. Low Count Radioactive Source Searching with Gaussian Process and Reinforcement Learning. IEEE *NSS-MIC*. Manchester, UK (2019)