SANDIA REPORT
SAND2024-02144
Printed February 2024

# Controlling Radioisotope Proportions when Randomly Sampling from Dirichlet Distributions in PyRIID

Alan J. Van Omen & Tyler J. Morrow

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

**NOTICE:** This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

| | |
|---|---|
| Telephone: | (865) 576-8401 |
| Facsimile: | (865) 576-5728 |
| E-Mail: | reports@osti.gov |
| Online ordering: | http://www.osti.gov/scitech |

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

| | |
|---|---|
| Telephone: | (800) 553-6847 |
| Facsimile: | (703) 605-6900 |
| E-Mail: | orders@ntis.gov |
| Online order: | https://classic.ntis.gov/help/order-methods |

## ABSTRACT

As machine learning models for radioisotope quantification become more powerful, likewise the need for high-quality synthetic training data grows as well. For problem spaces that involve estimating the relative isotopic proportions of various sources in gamma spectra it is necessary to generate training data that accurately represents the variance of proportions encountered. In this report, we aim to provide guidance on how to target a desired variance of proportions which are randomly when using the PyRIID Seed Mixer, which samples from a Dirichlet distribution. We provide a method for properly parameterizing the Dirichlet distribution in order to maintain a constant variance across an arbitrary number of dimensions, where each dimension represents a distinct source template being mixed. We demonstrate that our method successfully parameterizes the Dirichlet distribution to target a specific variance of proportions, provided that several conditions are met. This allows us to follow a principled technique for controlling how random mixture proportions are generated which are then used downstream in the synthesis process to produce the final, noisy gamma spectra.

This page intentionally left blank.

## ACKNOWLEDGMENT

This work was funded by the U.S. Department of Energy, National Nuclear Security Administration, Office of Defense Nuclear Nonproliferation Research and Development (DNN R&D).

This page intentionally left blank.

# CONTENTS

This page intentionally left blank.

## LIST OF FIGURES

This page intentionally left blank.

# ACRONYMS & DEFINITIONS

**ML** Machine Learning

**PyRIID** Sandia's Open-Source Python package of RIID-related software utilities

**RIID** Radioisotope Identification

This page intentionally left blank.

# 1. INTRODUCTION

When synthesizing gamma spectra containing counts from many radioisotopes at once, generating all possible combinations quickly becomes infeasible as mixture size grows. An alternative to enumeration is to randomly sample the target space an arbitrary number of times for arbitrary mixture sizes while ensuring proportions sum to one, and the Dirichlet distribution is convenient for this purpose. However, defining a problem space, especially as it pertains to parameterizing the Dirichlet distribution, remains a challenge. To help alleviate this challenge, this report aims to establish a principled methodology that can be readily employed with existing tools, namely PyRIID [1].

The practical offering of this report is a guide for PyRIID users on how to set the alpha parameter of the Dirichlet distribution when mixing spectral templates, or *seeds*. The foundation of this guidance pertains to selecting sparsity in order to control the shape of distributions of proportions, which we consider a more intuitive endeavor relative to selecting alpha, especially with the resources provided. We then go one step further to demonstrate how to apply this work in two case studies mirroring real-world scenarios where a priori proportion information is either known (from analyzing historical data, such as of fission sources) or unknown (most common).

This report is organized as follows:

- Section 2, Background, briefly introduces the concepts of the Dirichlet distribution, Gini Index, sparsity, and how all of it is used in PyRIID for synthesizing mixed gamma spectra.
- Section 3, Method, establishes the formal method by which sparsity can be targeted, how to select sparsity, and what the limitations are.
- Section 4, Case Studies, describes two ways in which our method can be employed.
- Section 5, Conclusion, summarizes the method.

This page intentionally left blank.

# 2. BACKGROUND

## 2.1. Dirichlet Distribution

The Dirichlet distribution, which is sometimes referred to as the multivariate beta distribution, is a continuous multivariate probability distribution commonly used to define a prior distribution of variables in Bayesian mixture modeling. It is commonly used in techniques related to natural language processing, such as the Dirichlet process and latent Dirichlet allocation [2, 3]. For PyRIID and mixing gamma spectrum seeds in random proportions, the Dirichlet distribution is useful because randomly sampled vectors (where the length of the vector is the target mixture size) have the useful property of always being proportions which sum to one. In this section we introduce and formally define the Dirichlet distribution. We also provide some intuition for understanding the Dirichlet distribution by providing an example and comparing it to its discrete versions.

### *2.1.1. Definition*

The Dirichlet distribution is a continuous multivariate probability distribution for $N$ categories, $\mathbf{x} = \{x_1, \ldots, x_N\}$, which is parameterized by a vector of $N$ parameters, $\alpha = \{\alpha_1, \ldots, \alpha_N\}$, where $\alpha_i > 0$. The PDF of the Dirichlet distribution is defined as,

$$P(\mathbf{x}; \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{N} x_i^{\alpha_i - 1},$$

where $B(\alpha) = \frac{\prod_{i=1}^{N} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{N} \alpha_i)}$ and $\Gamma(\cdot)$ is the gamma function.

The Dirichlet distribution has support consisting of N-dimensional vectors whose elements are positive and sum to one: $x_i \in [0, 1]$ and $\sum_{i=1}^{N} x_i = 1$. In other words, the Dirichlet distribution has support over valid probability distributions, which is known as the $(N - 1)$ probability simplex ($\mathbf{x} \in \Delta^{N-1}$). For this reason the Dirichlet distribution can be thought of as a distribution of distributions.

The parameter $\alpha$ defines the expected proportions and variance (or sparsity) of the proportions from the Dirichlet distribution. In the case that $\alpha_1 = \alpha_2 = \cdots = \alpha_N$, all the proportions will have equal expectations at $1/N$, and random samples will have equal proportions on average. In the case that $\alpha_1 = 2, \alpha_2 = 1, \ldots, \alpha_N = 1$, random samples would have a skewed shape, with the first proportion being higher on average. While the relative values of $\alpha_i$ controls the expectation of the proportions, the magnitude of $\alpha$ controls the variances. For example, as $\alpha_1 = \alpha_2 = \cdots = \alpha_N \to \infty$, the variance of randomly sampled proportions will decrease, converging on all equal proportions. As $\alpha_1 = \alpha_2 = \cdots = \alpha_N \to 0$, the variance of randomly sampled proportion will increase, converging on a one-hot vector with a single positive proportion equal to one.

### 2.1.2. *Intuition*

One intuitive application of the Dirichlet distribution would be a string partitioning application. Suppose that several pieces of string (all equal in length) were to be cut into $N$ pieces of varying lengths. The Dirichlet distribution can be used to model the lengths of the string partitions, and by changing $\alpha$, one can control the variation and expectation of the partition lengths. Figure 2-1 shows what some randomly sampled string lengths look like for $N = 3$. In particular, this demonstrates that the expected length for each partition is given by $\frac{\alpha_i}{\alpha_0}$, where $\alpha_0 = \sum_{i=1}^{N} \alpha_i$.

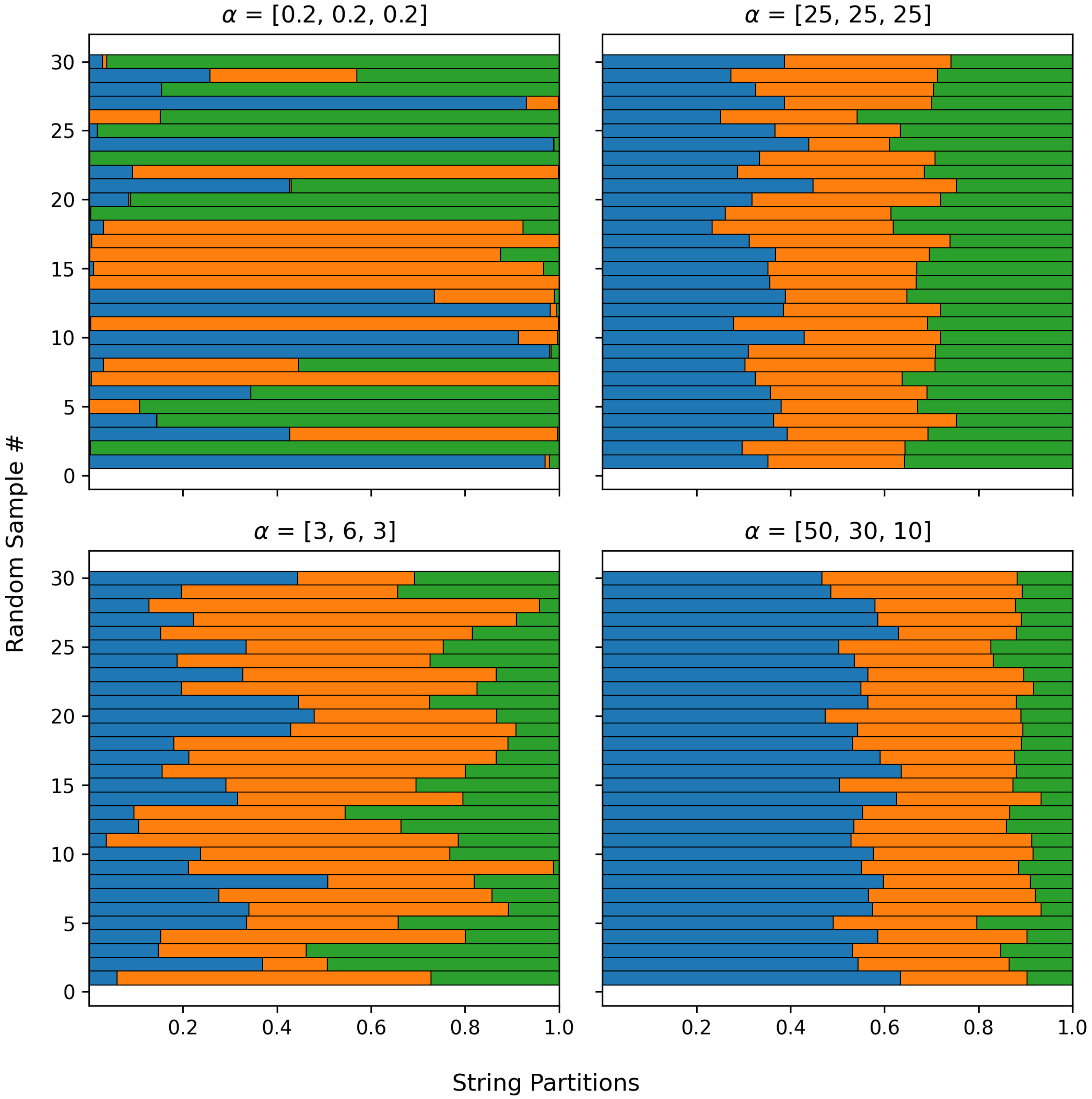

**Figure 2-1. Example of strings partitioned into $N = 3$ segments, with the segment lengths being sampled from various Dirichlet distributions**

Additional intuition for the Dirichlet distribution may be gleaned from comparing it the binomial, multinomial, and beta distributions. The binomial distribution defines the discrete probability distribution of observing $x$ "successes" out of $n$ trials, where each trial has only two outcomes - "success" or "failure" (for example, the number of heads out of $n$ coin tosses). These trials are known as Bernoulli trials. The PMF of the binomial distribution is defined as,

$$P(x; p, n) = \binom{n}{x} p^x (1-p)^{n-x},$$

where $p$ is the probability of a success. This formula has a very straightforward interpretation, where $\binom{n}{x}$ is the number of combinations with $x$ successes and $p^x(1-p)^{n-x}$ is the probability of observing $x$ successes and $n-x$ failures.

The multinomial distribution can be thought of as the generalized version of the binomial distribution for larger dimensions. Instead of a single scalar outcome, each trial has $N$ outcomes, each with their own success probability, defined as $\mathbf{p} = \{p_1, \dots, p_N\}$. The multinomial distribution is a multivariate discrete distribution which defines the probability of observing a set of successes for each outcome, defined as $\mathbf{x} = \{x_1, \dots, x_N\}$. The PMF of the multinomial distribution is defined as,

$$P(\mathbf{x}; \mathbf{p}, n) = \frac{n!}{\prod_{i=1}^{N} x_i!} \prod_{i=1}^{N} p_i^{x_i}.$$

The beta distribution can be thought of as a continuous version of the binomial distribution and is used for randomly sampling proportions, $x \in [0, 1]$. In particular, the beta distribution is a continuous probability distribution defined on the interval $[0, 1]$ using two positive scalar parameters, $\alpha$ and $\beta$. The PDF of the beta distribution is defined as,

$$P(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)},$$

where $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ and $\Gamma(\cdot)$ is the gamma function.

In the same way that the multinomial distribution is the generalized version of the binomial distribution, the Dirichlet distribution is the generalized version of the beta distribution.

### 2.2. Sparsity

The concept of sparsity is widely used in signal processing methods, and is often employed as an underlying assumption or is enforced through some constraint. Sparsity forms the basis of many techniques such as compressed sensing (CS) [4, 5], dictionary learning [6, 7, 8], signal recovery [9, 10], and matrix factorization [11]. It has been shown to be a powerful tool in various machine learning methods [12, 13, 14, 15], and is often leveraged in signal denoising [16, 17] and computer vision applications [18, 19].

Per [20], despite its prevalence in literature, there is no universal definition of sparsity, and it can be measured through a myriad of different metrics. In general, they say that a signal is sparse if a large proportion of its energy is contained in a few of its coefficients. Realizing this concept mathematically has led to numerous definitions of sparsity, however definitions generally share two key concepts:

1. a signal or distribution achieves maximum sparsity when all of its energy is contained in one coefficient; and
2. a signal or distribution has minimal sparsity when all its energy is evenly distributed.

In some literature, the $\ell^p$-norms are used to measure sparsity [5, 21, 22]. For a vector $\mathbf{x} \in \mathbb{R}^N$, these are defined as,

$$||\mathbf{x}||_p = \left( \sum_{i=1}^{N} x_i^p \right)^{1/p} .$$

The $\ell^0$-norm is traditionally used to measure sparsity in theoretical settings, and is defined as the number of non-zero coefficients. However, this definition of sparsity is often impractical as it cannot be used for optimization (its derivative does not change) and is meaningless in the presence of noise. As such, the $\ell^0$-norm is often approximated with $\ell^p$-norms where $0 < p \leq 1$. And the $\ell^1$-norm is often used as a convex relaxation of the $\ell^0$-norm and forms the basis for the popular LASSO technique [23, 24].

Hurley and Rickard in [20] compare 16 different measures for sparsity in terms of 6 mathematical criteria, the most important criteria being the "Robin Hood" principle (stealing from the rich to give to the poor should decrease sparsity). In their paper they demonstrate that only two sparsity metrics, the $pq$-mean with $p \leq 1, q > 1$ and the Gini index, satisfy all six properties. For this paper, we study the Gini index, which unlike the $pq$-mean, has the additional properties of being bounded between 0 (inclusive) and 1 (exclusive) and overall having less parameters on which to decide.

### 2.3. Gini Index

The Gini index, also known as the Gini coefficient or Gini ratio, is a measure of sparsity which was originally proposed as a metric of wealth inequality [25]. The Gini index has been successfully used as a sparsity measure in various applications [26, 27], and is commonly employed as an impurity measure in decision tree algorithms, such as a random forests [28, 29].

For vector $\mathbf{x} \in \mathbb{R}^N$ sorted in a non-decreasing order, $x_i \leq x_{i+1}$, the Gini index is defined as,

$$G(\mathbf{x}) = 1 - 2 \sum_{i=1}^{N} \frac{x_i}{||\mathbf{x}||_1} \left( \frac{N - i + \frac{1}{2}}{N} \right).$$

For any $\mathbf{x} \in \mathbb{R}^N$ with non-negative values the Gini index is bounded on $G(\mathbf{x}) \in [0, 1)$. A Gini index of 0 indicates minimum sparsity, where the given distribution has perfect equality (all the components are equal). And for the case of maximum sparsity, where there is only one non-zero component with all the signal energy, $G(\mathbf{x}) = 1 - 1/N$, which we find inconvenient and modify in Section 3.1.

The Gini index also has an intuitive graphical interpretation, shown in Figure 2-2. The Lorenz curve for a distribution is a line showing the cumulative coefficient value versus the percentage of coefficients. The Gini index is defined to be exactly twice the area between the Lorenz curve and the 45° line, which represents a distribution with perfect equality.

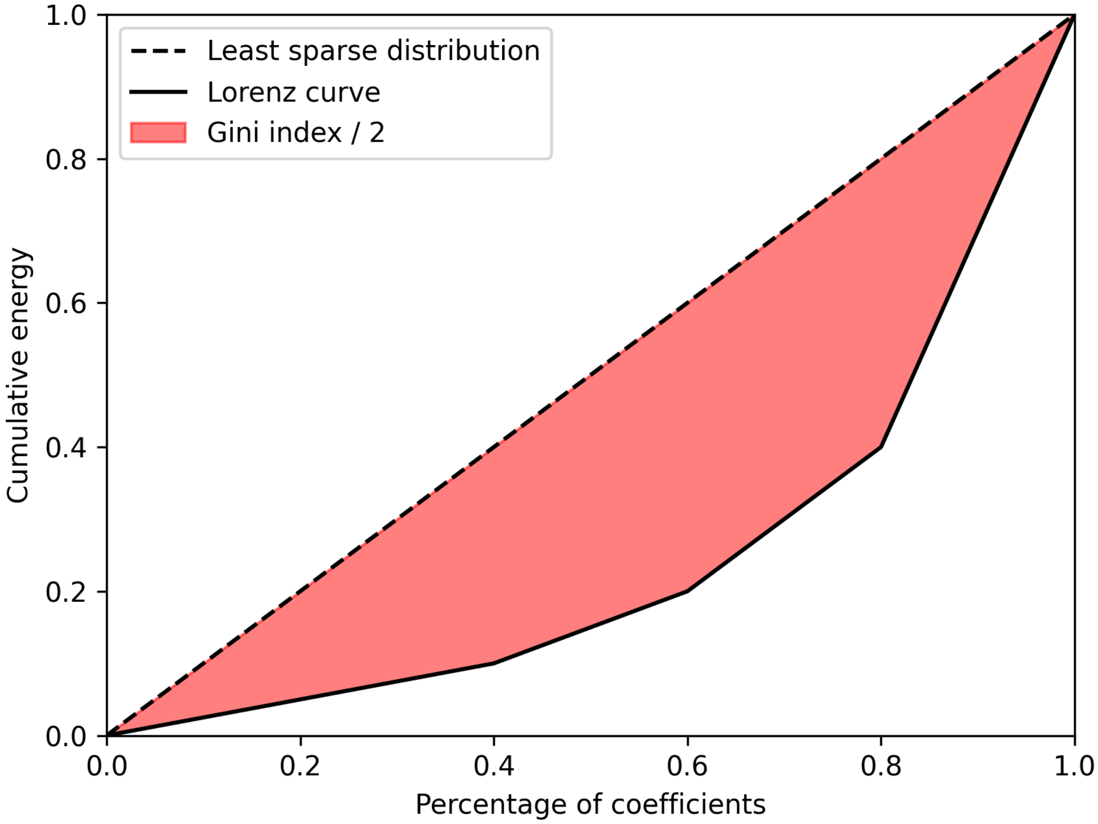

**Figure 2-2. A Lorenz curve for $\mathbf{x} = [0.05, 0.05, 0.1, 0.2, 0.6]$, the Gini index is equal to twice the shaded area**

## 2.4. PyRIID

PyRIID (pronounced: PIE-rid) is a Python package that facilitates gamma spectrum synthesis, model training, and visualizations for machine learning-based (ML) radioisotope identification (RIID) [1]. Data synthesis occurs in up to three stages:

1. Seed Synthesis: the first step is to obtain *seed* spectra (seeds are source templates without Poisson noise) to which to fit, or on which to test, a model. PyRIID does this by providing

a Python wrapper around the GADRAS API and accessing its Inject capabilities to produce seeds for a specific Detector Response Function (DRF) [30]. While assumptions about sources, detector, and environment are all made here, the ability to vary any parameter is provided. From this stage, the user will end up with two sets of seeds: those intended as anomalous sources and those intended as backgrounds.

2. Seed Mixing: the obtained seeds are then summed together in proportions randomly generated from a Dirichlet distribution. The most common use of the mixer is to construct a variety of background samples from seeds of K, U, T, and Cosmic. This step is the focus of this report as we want to control the formation of datasets features mixtures both large and small. In addition, mixed spectra are often obtained at the seed synthesis stage via GADRAS's modeling functionality to establish geometric and activity ratios. When doing this, it is important to understand that at the time of writing, the precise composition of the mixtures in terms of count contribution will be lost. Such pre-mixed seeds can be used in the seed mixing, but extra attention must be given to understand what subsequent proportion estimators or algorithms will do with such data.

3. Static Synthesis: seeds, or mixed versions of them, are then randomly varied in terms of signal-to-noise ratio (SNR) and live time for an expected background rate to obtain gross spectra. Poisson noise can then be applied.

Throughout the synthesis process, careful ground truth tracking is performed to preserve the proportions by which sources contribute to the counts in a sample. Final, noisy samples are often then taken to fit supervised or unsupervised classifiers or estimators for useful tasks, or simply just to test existing models and algorithms.

# 3. METHOD

When generating synthetic training data of mixed gamma spectra with PyRIID [1], the SeedMixer samples random isotope proportions using the Dirichlet distribution, as discussed in Section 2.4. Section 2.1.1 shows how the distributional shape and variance of these random isotope proportions are a function of the $\alpha$ parameter. Properly setting $\alpha$ is crucial to generating useful training data. Setting $\alpha$ too large will cause all the isotope proportions to concentrate around their expectations, and thus a model will less likely to generalize to the data. If $\alpha$ is too small then all the isotope proportions will be either 0 or 1, and thus a model will not be exposed to mixtures at all. As such, the question of how to set $\alpha$ for a given problem came up frequently for the authors and developing a principled approach for setting $\alpha$ was needed. This section provides an intuitive method for setting $\alpha$ using a measure of sparsity based on a modified version of the Gini index.

## 3.1. Regularized Gini Index

In this section we define a custom measure of sparsity which is based on the Gini index (Section 2.3). For an N-dimensional vector of isotope proportions, $\mathbf{x} \in \Delta^{N-1}$, we want our sparsity measure, $S(\mathbf{x})$, to be easily interpretable with $S(\mathbf{x}) = 0$ indicating minimum sparsity and $S(\mathbf{x}) = 1$ indicating maximum sparsity. In particular, we want $S(\mathbf{x})$ to satisfy the following criteria:

1. Our sparsity measure must be smooth and bounded by $S(\mathbf{x}) \in [0, 1]$.
2. When all the isotope proportions are exactly equal to their expectation, $S(\mathbf{x}) = 0$ (note that here this intentionally overlaps with the case of isotope proportions being equal).
3. When there is only one positive isotope proportion equal to 1 (i.e., $\mathbf{x}$ is a one-hot vector), $S(\mathbf{x}) = 1$.

The Gini index seems ideally suited for this application except for two caveats.

- First, a maximally sparse vector has a Gini index of $G(\mathbf{x}) = 1 - 1/N \neq 1$, thus failing to meet the first and third criteria. In order to meet these criteria, we can simply scale the Gini index by $\frac{N}{N-1}$. Notice that $\frac{N}{N-1}(1 - 1/N) = 1$.
- Secondly, when isotope proportion expectations are not uniform, $G(E[\mathbf{x}]) > 0$, thus failing to meet the second criterion. We can account for this by simply dividing each proportion by its corresponding expectation before computing the Gini index. In particular, $G(\mathbf{x} \oslash E[\mathbf{x}]) = G(E[\mathbf{x}] \oslash E[\mathbf{x}]) = G(\mathbf{1}_N) = 0.0$, thus satisfying the second criteria.

Thus, for an N-dimensional vector of isotope proportions $\mathbf{x} \in \Delta^{N-1}$ and a vector of expectations $E[\mathbf{x}] \in \Delta^{N-1}$, we define our sparsity measure $S(\mathbf{x}; E[\mathbf{x}])$ as a regularized version of the Gini index,

$$S(\mathbf{x}; E[\mathbf{x}]) = \frac{N}{N-1} G(\mathbf{x} \oslash E[\mathbf{x}]), \tag{3.1}$$

where $G(\cdot)$ denotes the Gini index and $\oslash$ represents the Hadamard division operator (element-wise division).

### 3.2. $\alpha$ Determination

This section shows how $\alpha$ should be selected in order to randomly sample isotope proportions for a given expectation, $E[\mathbf{x}] \in \Delta^{N-1}$, and sparsity level, $S \in [0, 1]$. In order to target a particular distribution of expectations, the Dirichlet generator must be parameterized by an $\alpha$ whose elements are proportional to the given expectation. In particular, $\alpha$ must take the following form,

$$\alpha = \lambda * E[\mathbf{x}],$$

where the relative weights in $\alpha$ are already set, and only a positive multiplier $\lambda \in \mathbb{R}^{+}$ can be adjusted.

Then all that is left is to determine some function which maps the provided expectation and sparsity level to a scalar multiplier, i.e. $\lambda = f(S, E[\mathbf{x}])$. Due to our perception of the infeasibility of deriving a closed form expression for the CDF of a Dirichlet distribution, we instead relate the Gini index to $\alpha$ via a lookup function, $f$. This lookup is built through empirically quantifying the relationship between $\lambda$ and $S, E[\mathbf{x}]$ using Monte Carlo simulation.

For the case where each isotope proportion has an equal expectation (i.e., $E[\mathbf{x}] = [1/N, \ldots, 1/N]$), Figure 3-1 shows the sparsity level in terms of the mixture size $N$ and $\lambda$. For each combination of $N$ and $\lambda$, the sparsity level, $S$, is determined by drawing 1000 random samples from a Dirichlet distribution and computing the average. This figure shows that for sufficiently small $\lambda$, $S \longrightarrow 1.0$, and for sufficiently large $\lambda$, $S \longrightarrow 0.0$, as desired. Figure 3-1 also shows that the relationship between $\lambda$ and $S$ is dependent on the mixture size, $N$.

Figure 3-2 shows the sparsity level in terms of the mixture size $N$ and $\lambda$ for the case of non-uniform expectation. In this case, the expectations take the form of,

$$E[\mathbf{x}] = [0.9, \frac{0.1}{N-1}, \ldots, \frac{0.1}{N-1}].$$

From comparing Figs. 3-1 and 3-2 it is clear that the relationship between $\lambda$ and $S$ depends not only on the mixture size, $N$, but also on the distribution of the expectations themselves, $E[\mathbf{x}]$.

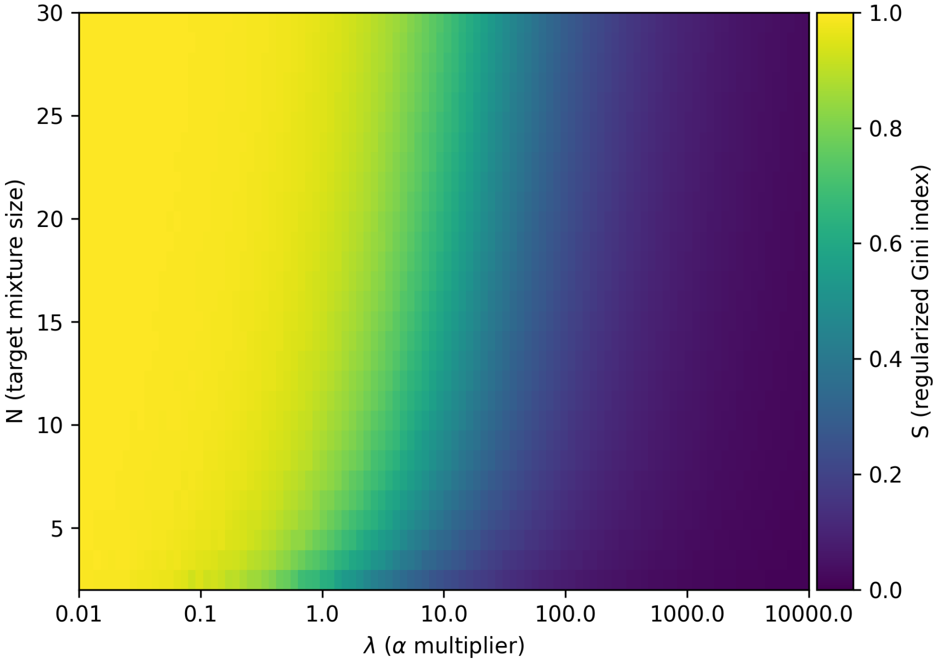

**Figure 3-1. Sparsity level as a function of $N$ and $\lambda$ for the case of uniform expectation**

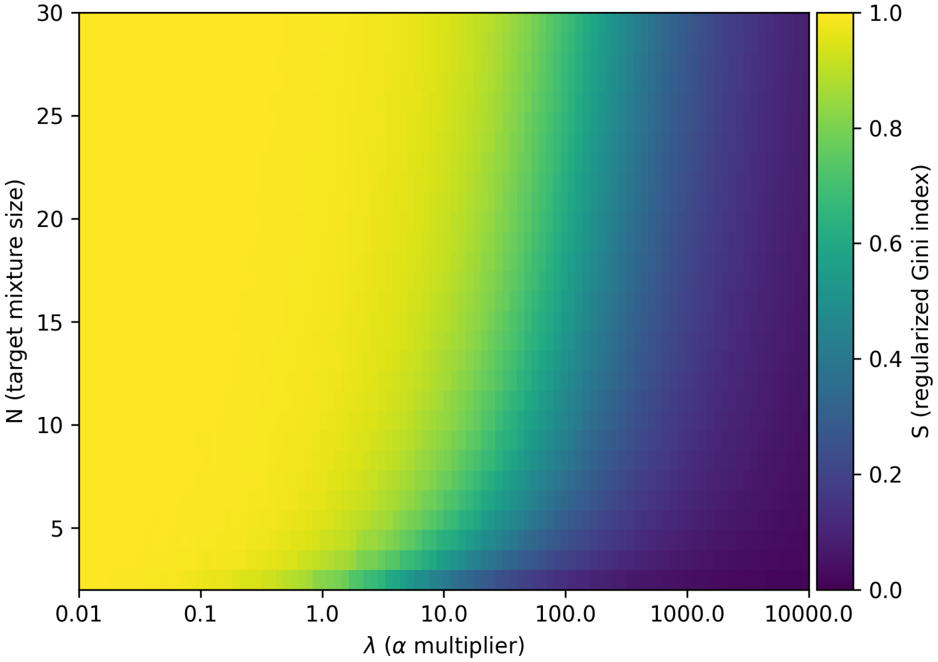

**Figure 3-2. Sparsity level as a function of $N$ and $\lambda$ for the case of non-uniform expectation where** $E[\mathbf{x}] = [0.9, 0.1/(N-1), \ldots, 0.1/(N-1)]$

With this in mind, we propose to empirically derive $f(S, E[\mathbf{x}])$ as an interpolated univariate spline function fit using Monte Carlo random sampling. In particular, we derive a unique lookup function for a given expectation, $E[\mathbf{x}]$, through the following steps.

1. For a range of possible $\lambda$s, generate a set number of i.i.d. samples from a Dirichlet distribution parameterized by $\alpha = \lambda * E[\mathbf{x}]$. In our case, we sample 25 $\lambda$s on a log-distribution ranging from 0.01 to 10000. For each $\lambda$ we draw 100k random samples from the associated Dirichlet distribution.

2. Compute the mean sparsity level, $S$, using Equation 3.1 for the random samples generated using each $\lambda$.

3. Using the mean sparsity levels as x-values and the $\lambda$s as y-values, fit a smooth function. In our case, we selected an interpolated univariate spline function implemented in Scikit-learn [31].

Figure 3-3 shows the lookup function, $f(S, E[\mathbf{x}])$, generated for an example expectation $E[\mathbf{x}] = [0.3, 0.2, 0.5]$. For a targeted sparsity level of $S = 0.5$, this function returns the following $\lambda$,

$$\lambda = f(S = 0.5; E[\mathbf{x}] = [0.3, 0.2, 0.5]) \approx 3.391.$$

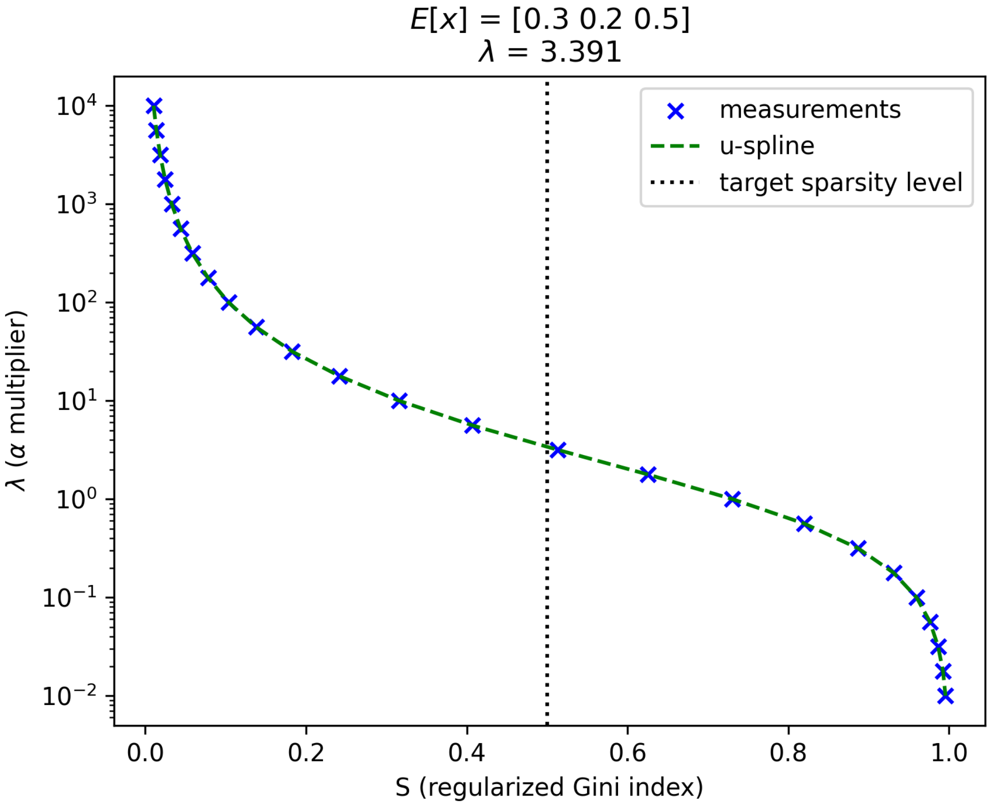

**Figure 3-3. Example of generated lookup function $f(S; E[\mathbf{x}])$ for $E[\mathbf{x}] = [0.3, 0.2, 0.5]$, shows for a target sparsity level of $S = 0.5$ that $\lambda \approx 3.39$**

### 3.3. Sparsity Determination

Section 3.2 provides a method for parameterizing the Dirichlet distribution in order to target a specific sparsity level. This section provides guidance on how to select an appropriate sparsity level, $S$.

In the most ideal setting, one would have access to a sufficiently large set of labeled isotope proportions from a test scenario to directly estimate $S$. In this case, $S$ could be empirically measured for all the test samples and the average would be set as the input to $f$. Or better yet, one could directly estimate $\alpha$ for the Dirichlet distribution using some maximum likelihood estimation (MLE) technique such as those detailed in [32, 33, 34].

However, a sufficiently large number of labeled test proportions is rarely available, in which case a sparsity level must be selected based on some other criteria. As the point of this paper is to obtain mixtures, one should avoid picking a sparsity level, $S$, close to its extremes (0 or 1).

- As $S \longrightarrow 0$, the sparsity of the generated proportions will be minimal and all the proportions will converge on exactly their expectation. In this case there will be no variation in training proportions and a model, having no useful information to generalize with, will simply converge on predicting the expectation.
- As $S \longrightarrow 1$, the sparsity of the generated proportions will be maximized and each sample will only have a single positive proportion equal to 1.0. In this case the training data represents a classification problem, and in the context of PyRIID, mixing would not be needed anyway.

Therefore, a useful sparsity level to target should always be "somewhere in the middle." In other words, $S$ should be small enough so that training data accurately models the expectation, but large enough so that the model does not simply learn to predict the expectation.

With this in mind, we propose to select a sparsity level, $S$, based on the *effective* mixture size of the generated proportions. We define the effective mixture size, $N_\epsilon$, to be the number of isotope proportions $> \epsilon$. By this definition, the effective mixture size is closely related to sparsity, which can be thought of as the case when a large amount of energy is represented with few coefficients. Along these lines, the effective mixture size is the number of significant contributors in a gamma spectrum. Note that $\epsilon$ could be selected based on application-specific criteria, such as the minimal detectable proportion of an isotope. In particular, when sampling random proportions will very low sparsity, the generated proportions should all be very close to their expectation, and thus $N_\epsilon \longrightarrow N$. However, when the sparsity is very high, the proportions should become one-hot vectors and $N_\epsilon \longrightarrow 1$. Figure 3-4 shows the mode of effective mixture sizes, $N_{0.01}$, for a range of mixture sizes, $N$, and sparsity levels, $S$, given a uniform expectation.

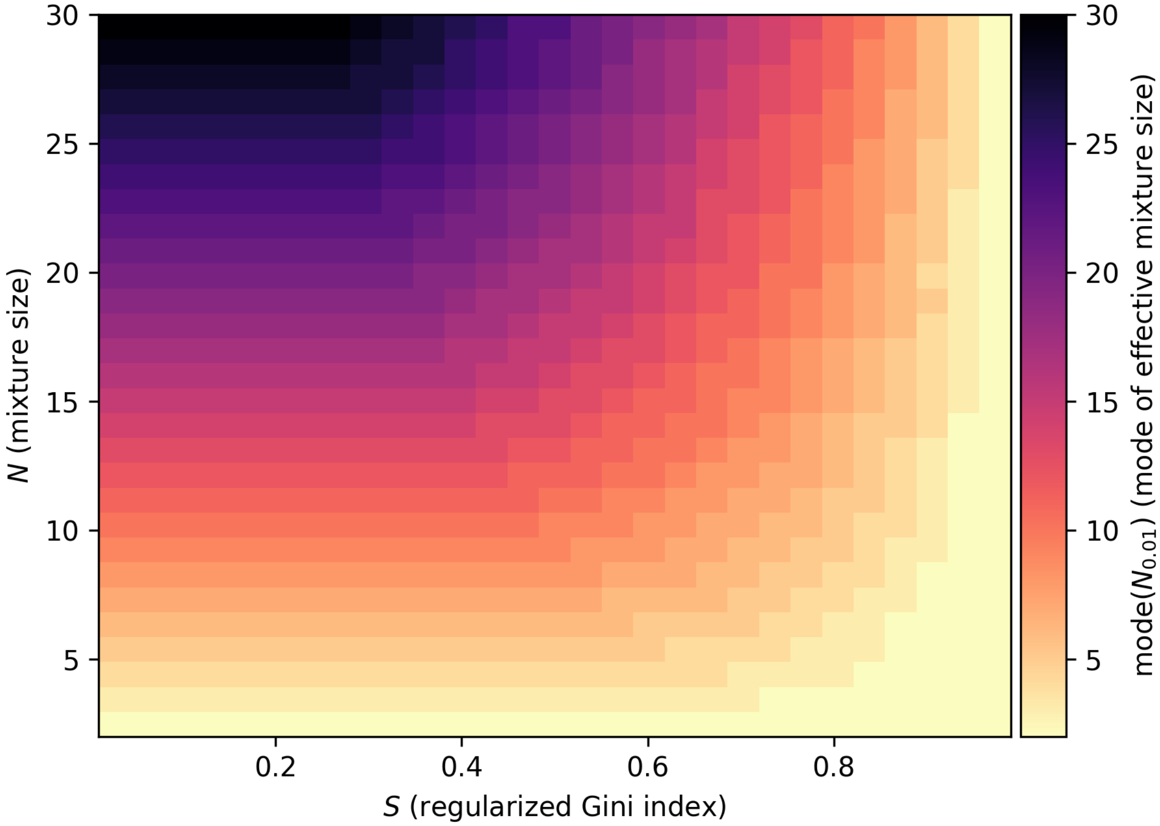

**Figure 3-4. Mode of effective mixture size, $N_{0.01}$, as a function of mixture size, $N$, and target sparsity level, $S$**

As an example, consider the case where $N = 10$ and $E[\mathbf{x}] = [0.1, \ldots, 0.1]$. Define the effective mixture size of a spectrum to be the number of proportions greater than 1% ($\epsilon = 0.01$). For this case, Figure 3-5 shows the distribution of effective mixture sizes for various levels of sparsity. Suppose we anticipate encountering a spectrum with around 8 sources simultaneously in our test environment. Then based on Figure 3-5, we should target a sparsity level of $S \approx 0.6$.

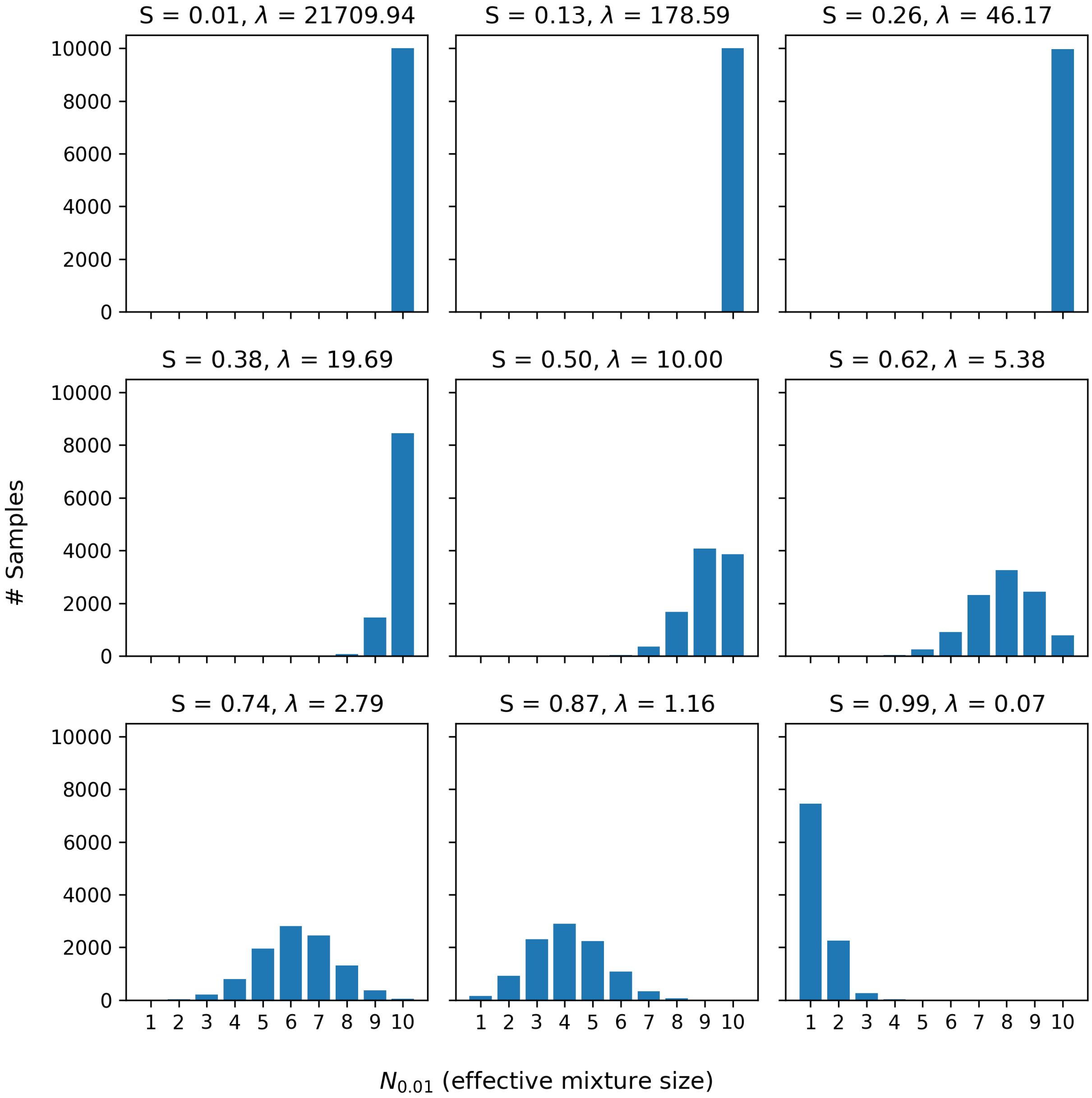

**Figure 3-5. Distribution of $N_{0.01}$ for various targeted sparsity levels for $N = 10$ with uniform expectation**

### 3.3.1. Targeting a Range of Sparsity (Optional)

In the case where no information is known regarding the expected mixture size, it may be desirable to target multiple sparsity levels a wide range of effective mixture sizes are present in the training data. For example, for the case shown in Figure 3-5, we can intuitively see that by combining proportions generated from various sparsity levels could more evenly represent the range of possible effective mixture sizes. In this section, we demonstrate how a linear combination of proportions from various sparsity levels can be set to target any distribution of effective mixture sizes.

In particular suppose we want each effective mixture size to have the same number of samples. To achieve this we must combine sampled proportions generated from a combination of different sparsity levels. In particular, we want to know what linear combination of the effective mixture size combinations (such as those shown in Figure 3-5) should be used to create a uniform distribution. We can express this problem as a linear matrix equation and determine the linear combination with the closest solution (in terms of the Euclidean distance) via least-squares minimization.

Suppose we have a set of distributions for $N_\epsilon$ (such as those shown in 3-5). Define $\mathbf{A} \in \mathbb{R}^{N \times T}$ to be a matrix whose columns are the vectors with the normalized shapes of those distributions, where $T$ is the number of sparsity levels considered. Define $\mathbf{b} \in \mathbb{R}^N$ to be a vector of the target distribution shape for $N_\epsilon$. Then the goal is to learn a vector of positive coefficients, $\mathbf{s} \in \mathbb{R}^T$, where each coefficient corresponds to the relative number of samples generated for that corresponding sparsity level. This can be expressed as the following linear matrix equation,

$$\mathbf{A}\mathbf{s} = \mathbf{b}. \tag{3.2}$$

We can solve Equation 3.2 using a non-negative least-squares (NNLS) optimizer such as the method implemented in Scipy [35].

As an example, consider the scenario where $N = 10$, $\epsilon = 0.01$, and $E[\mathbf{x}]$ is uniform (like the example from 3.3). Suppose we want to select multiple sparsity levels such that the final distribution of $N_{0.01}$ is approximately uniform (i.e., $\mathbf{b} = [0.1, \ldots, 0.1]$). We can achieve this using the aforementioned method by first generating a set of distributions of $N_{0.01}$ for a range of sparsity levels (like those shown in Figure 3-5). These distributions will be templates which comprise the columns of $\mathbf{A}$. Suppose we sample $T = 25$ sparsity levels, $S$, which are evenly distributed between 0.01 and 0.99. By solving for the coefficients, $\mathbf{s}$, using NNLS, we obtain a vector of relative coefficients for the number of samples that should be drawn from each sparsity level. Figure 3-6 shows the resulting distribution of $N_{0.01}$ for 100k samples drawn using this combination of sparsity levels. In practice, if a single sparsity level does not provide the desired distribution of training proportions, multiple sparsity levels can be combined in this way to target any provided distribution of $N_\epsilon$.

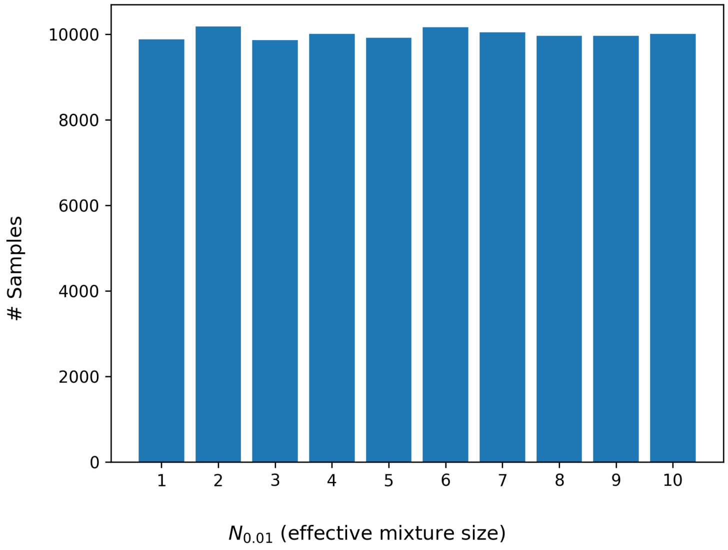

**Figure 3-6. Distribution of $N_{0.01}$ for with $N = 10$ and uniform expectation for a linear combination of various sparsity levels found using NNLS optimization.**

This page intentionally left blank.

# 4. CASE STUDIES

The purpose of this work is to provide a guide for achieving a desired level of sparsity when setting the $\alpha$ parameter of a Dirichlet distribution. The practical use of interest to the authors is mixing different gamma spectrum templates, such as with the SeedMixer in PyRIID [1]. This section performs two case studies to demonstrate how to use the $\alpha$ determination method (described in Section 3) in practice. Case study 1 looks at the scenario where expected proportions of mixtures are unknown, which is most common. Ideally, expected proportions could be determined a priori from analysis of historical data, but such information is often not available. As such, one might mix in an unbiased fashion until more real-world information becomes available. Case study 2 then looks at where proportions are known. When human analysis, often software-assisted, yields information about the expected isotopic composition, one could leverage such information to narrow the problem scope. On the hand, one may consider this intentionally biasing a model to a specific problem, but studying the trade-offs here is outside the scope of this paper and ultimately specific to the goals and consequences of the application.

## 4.1. Case Study 1: Unknown Mixture Proportions

In this case study we consider the scenario where we have minimal a priori information regarding our problem space. In particular, we consider the problem where we have 5 different isotopes for which we would like to estimate their relative isotopic proportions. We assume no access to test samples or any prior knowledge regarding their relative expectations. A real-world analogue to this case study would be background spectra which contain 5 components: Potassium, Uranium, Thorium, Cosmic, and intrinsic counts. Following the method defined in Section 3 we use the following steps to parameterize a Dirichlet distribution and generate mixed spectra.

1. The first step is to generate a lookup function, $f(S; E[\mathbf{x}])$ for our expectation. Because we have no prior knowledge regarding the expectation, we simply assume it is uniform:

$$E[\mathbf{x}] = [0.2, 0.2, 0.2, 0.2, 0.2].$$

   Then following the steps outlined in Section 3.2, we generate a lookup function which is shown in Figure 4-1.

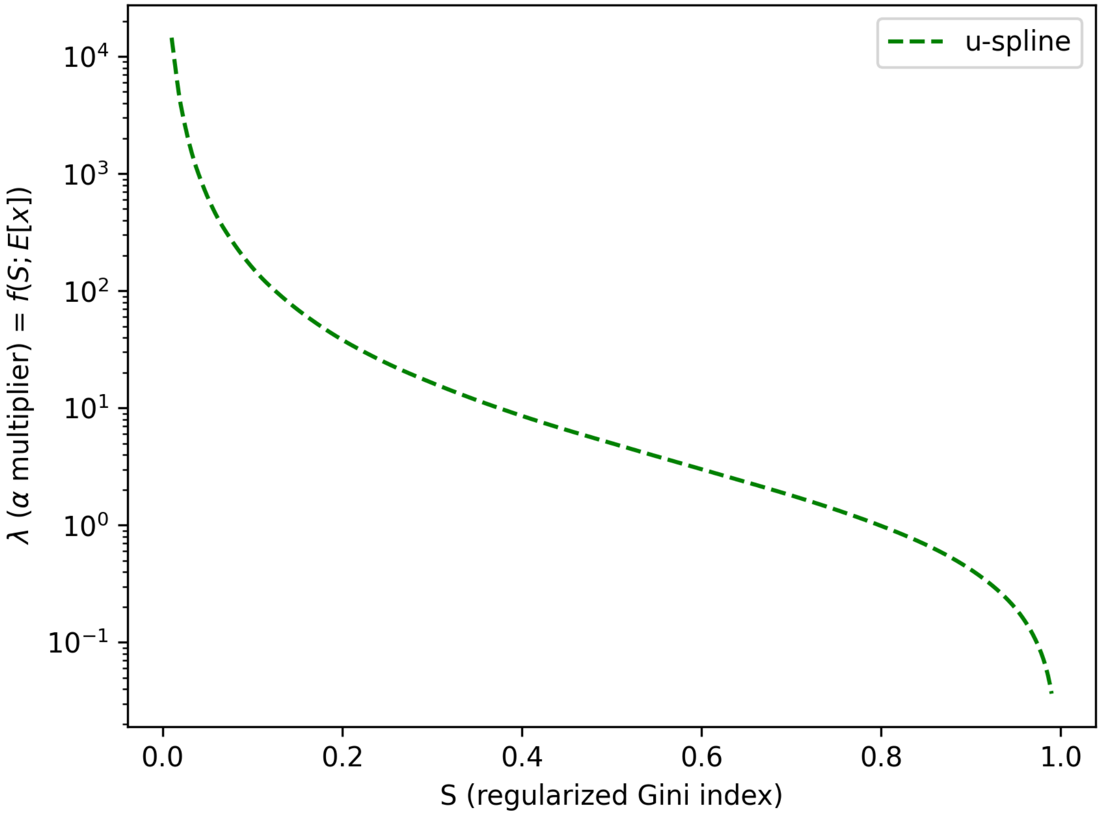

**Figure 4-1. Lookup function,** $f(S; E[\mathbf{x}])$**, for** $E[\mathbf{x}] = [0.2, 0.2, 0.2, 0.2, 0.2]$

2. With a generated lookup function, we can now visualize the distribution of effective mixtures sizes for a range of sparsity levels. Suppose we make the assumption that the minimal detectable proportion of an isotope is 5% of the counts. Figure 4-2 shows the effective mixture size distributions for sparsity levels ranging from 0.01 to 0.99.

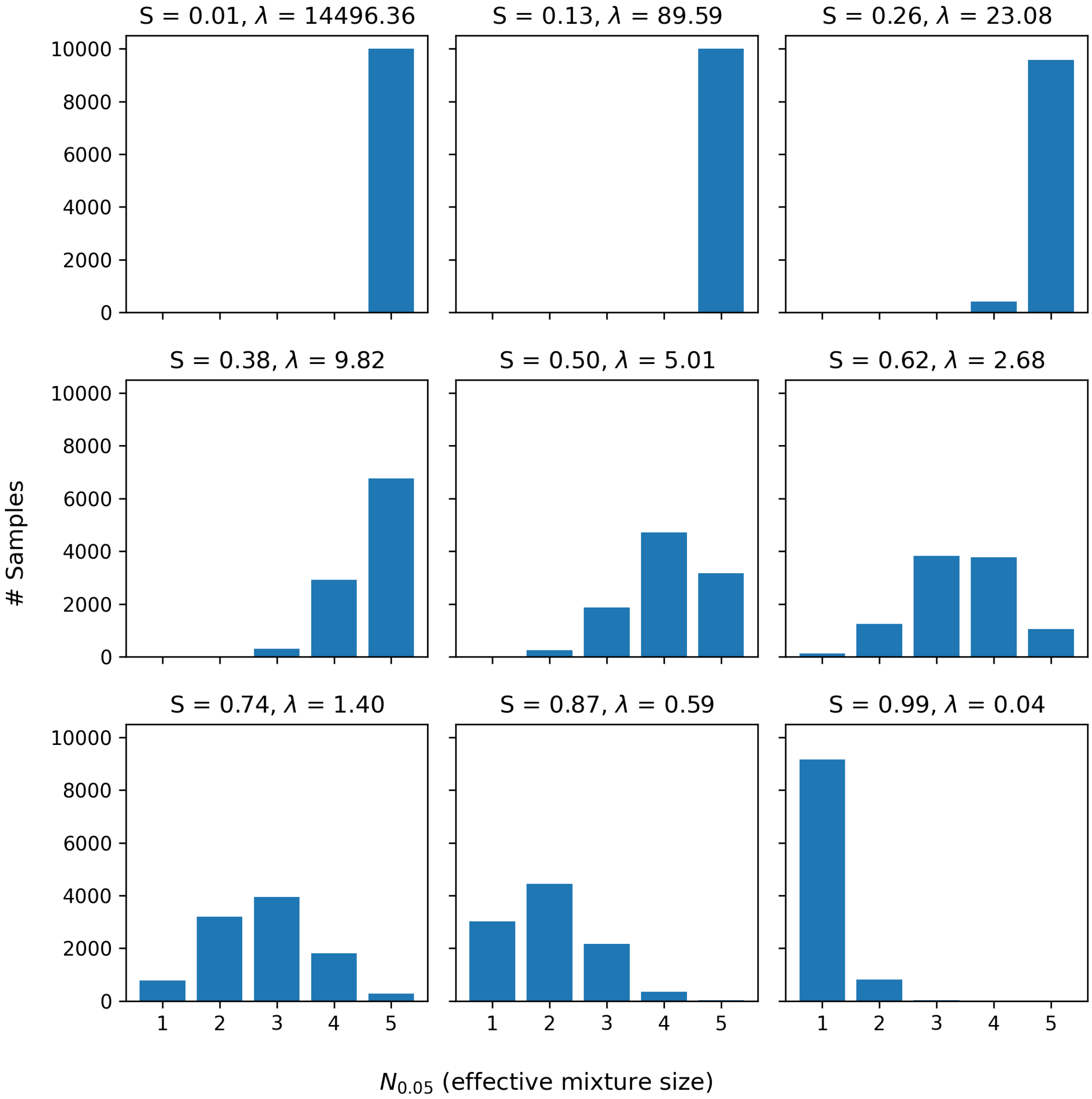

**Figure 4-2. Distributions of $N_{0.05}$ for various sparsity levels for case study 1**

For background mixtures, the number of contributors can typically be known beforehand, in which case that specific effective mixture can be targeted. On the other hand, if we have no knowledge regarding how many contributors we might expect to see simultaneously, a logical approach would be to balance the number of spectra containing each effective mixture size.

In this case, we would set $\mathbf{b} = [0.1, \ldots, 0.1]$. Figure 4-2 confirms that no single sparsity level will produce the desired distribution, so a linear combination of various sparsity levels was obtained using the method in Section 3.3.1. We select the 9 distributions shown in Figure 4-2 as templates, and after using NNLS to find the optimal linear combination of these templates, we obtain the following vector of coefficients,

$$\mathbf{s} = [0.0069,\ 0,\ 0,\ 0.2379,\ 0,\ 0.3187,\ 0,\ 0.3259,\ 0.1106].$$

3. An optional validation step was taken to make sure that the generated proportions have the correct distribution of $N_{0.05}$ and to ensure they yield an average sparsity matching the linear combination of target sparsity values. This was done by drawing a total of 100k random samples split between various sparsity levels by the coefficients found from the NNLS solution. The exact number of samples drawn from the 9 sparsity templates are shown in Table 4-1.

**Table 4-1. Number of samples drawn from each sparsity level for case study 1**

| Sparsity Level (S) | # Samples |
|---|---|
| 0.01 | 687 |
| 0.13 | 0 |
| 0.26 | 0 |
| 0.38 | 23788 |
| 0.5 | 0 |
| 0.62 | 31870 |
| 0.74 | 0 |
| 0.87 | 32594 |
| 0.99 | 11060 |

Figure 4-3 shows the distribution of $N_{0.05}$ for proportions generated from the linear combination of sparsity levels shown in Table 4-1. This demonstrates that the distribution of $N_{0.05}$ is approximately uniform as we desired. Our target sparsity, computed as the linear combination of the sparsity levels, was 0.6805. The average sparsity of the generated proportions was measured to be 0.6806. The distribution of the generated proportions is shown in Figure 4-4.

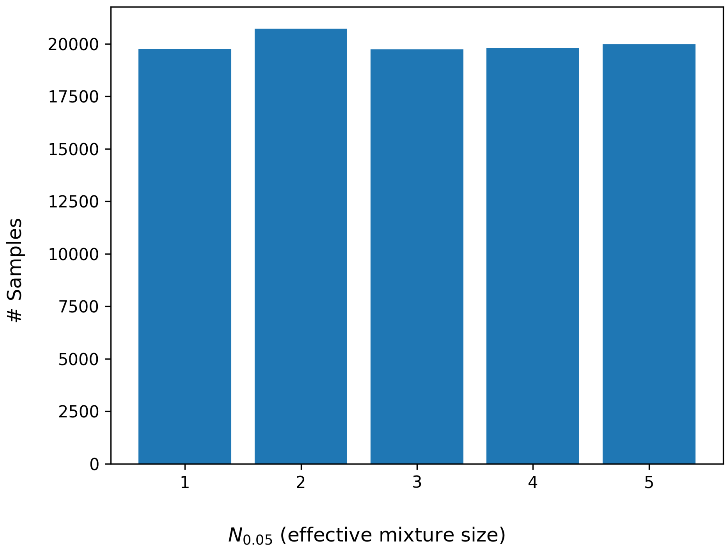

Figure 4-3. Distributions of $N_{0.05}$ for combination of sparsity levels for case study 1

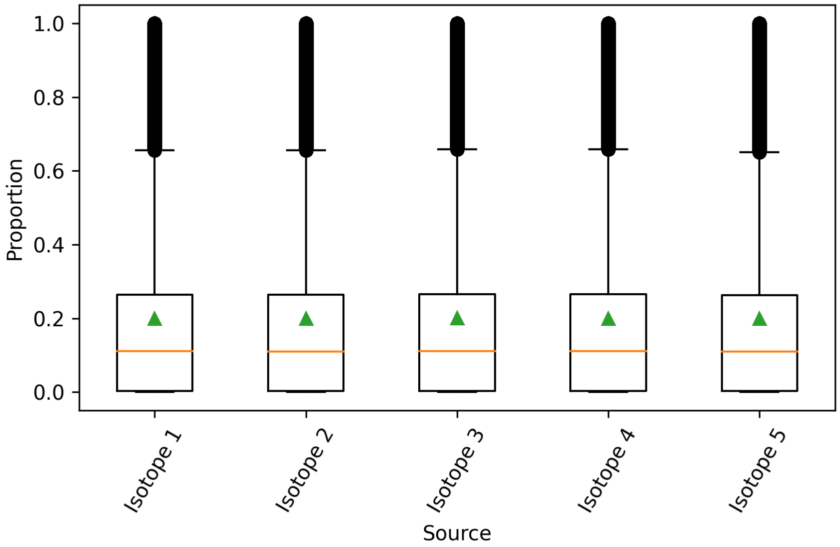

Figure 4-4. Box plot showing distribution of isotope proportions for case study 1

## 4.2. Case Study 2: Known Mixture Proportions

This case study considers a real-world application where more prior information is available. In particular here we apply this method to modeling gamma spectra emitted from spent nuclear fuel from a fission reaction as was done in [36]. In this case we have access to a real gamma spectrum which was obtained from [37]. The real gamma spectrum, measured some time after the reaction, was then analyzed by a spectroscopist and isotope proportion estimates were obtained for 30 distinct radioisotopes. The sources along with their proportions of counts contributed to the spectrum are shown in Figure 4-5.

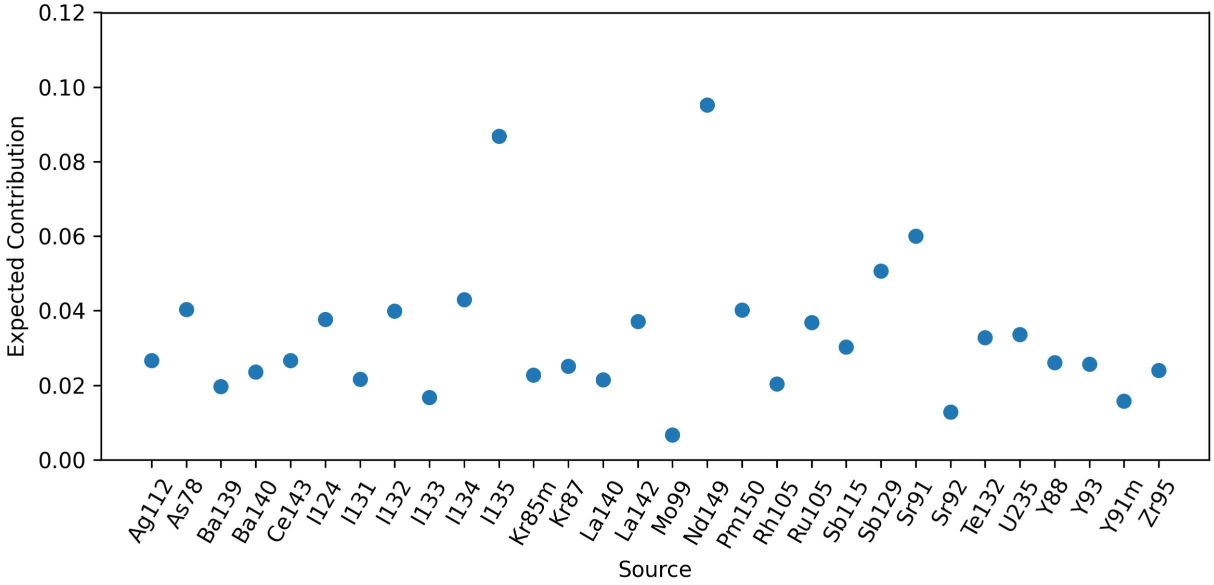

**Figure 4-5. Expected relative isotope proportions of spent nuclear fuel from a fission reaction for case study 2**

Using the same steps as in case study 1, we can generate data to model this scenario as follows.

1. The first step is to generate a lookup function for these particular expectations as detailed in Section 3.2. The lookup function for this particular case is shown in Figure 4-6.

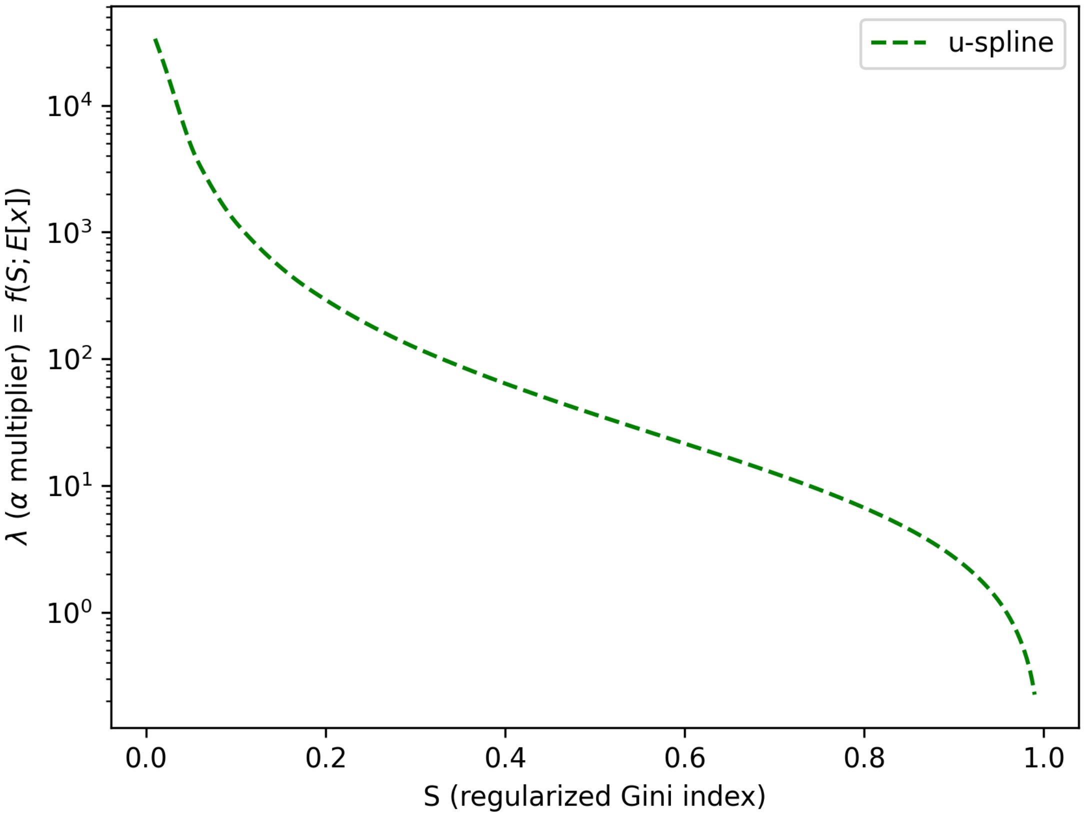

**Figure 4-6. Lookup function, $f(S; E[\mathbf{x}])$, generated for expectations obtained for SME analysis of fission source**

2. The next step is to select an appropriate sparsity level to target. Defining a significant contribution as proportions above 0.01, Figure 4-7 shows the distributions of effective mixture size for various sparsity levels. For this application, we are expecting to see spectra which have similar proportions to these expectations as a downstream algorithm might be designed to accurately estimate proportions, but only within a certain timeframe. In particular, the expectation from Figure 4-5 indicates that 29 out of the 30 sources have expected contributions above 1% of the total counts. Therefore, from Figure 4-7 we select $S = 0.15$ as this should yield proportions where the most common effective mixture size of 29. The generated lookup function yields a $\lambda$ of,

$$\lambda = f(S = 0.15; E[\mathbf{x}]) \approx 528.$$

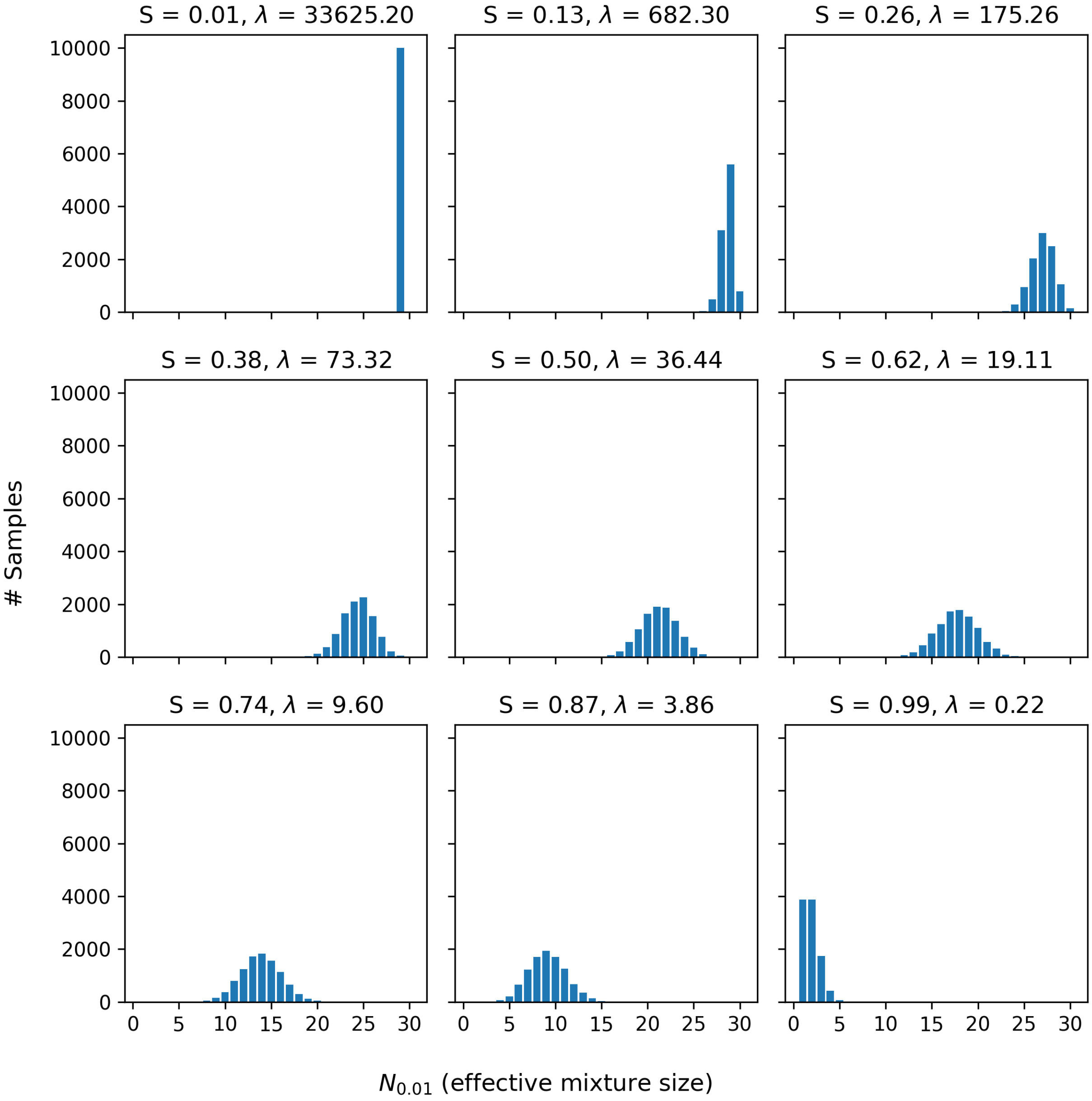

**Figure 4-7. Distributions of $N_{0.01}$ for various sparsity levels for case study 2**

3. After generating 100,000 samples with random isotope proportions targeting a sparsity of $S = 0.15$, the average measured sparsity was 0.150. Figure 4-8 shows the distribution of generated proportions for each isotope. The reader may note how the distribution of proportions for each isotope is closely concentrated around the expectation which makes sense given the low sparsity level. However, this dataset, collectively bearing low sparsity, may be difficult for trained models to generalize on. If the model converges on predicting the expectation for every isotope, consider adding spectra generated from higher sparsity levels and using calibration plots for verification (predicted proportion versus expected proportion).

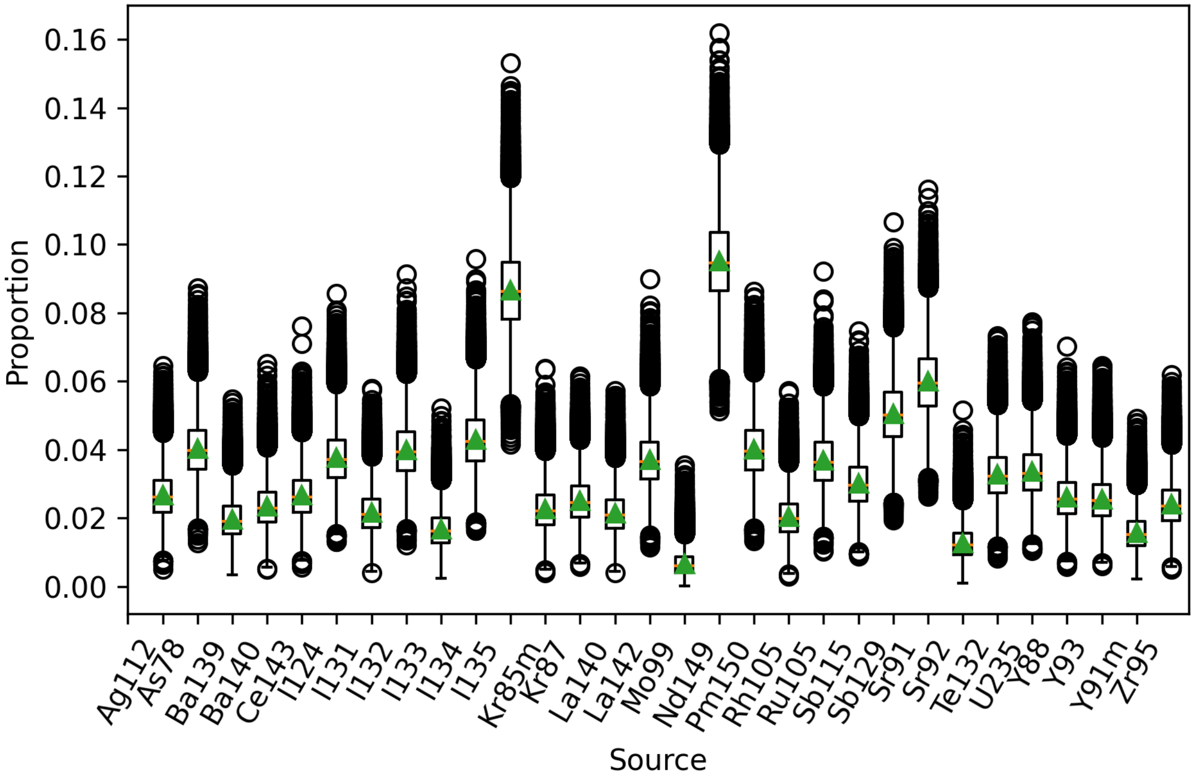

**Figure 4-8. Distributions of isotope proportions for case study 2**

This page intentionally left blank.

# 5. CONCLUSION

In conclusion, this study demonstrated how the Dirichlet distribution can be used to sample random mixture proportions based on a desired level of sparsity. The method presented helps set the $\alpha$ parameter of the Dirichlet distribution in order to target a specific degree of sparsity based on the Gini index. For sampling mixtures where a priori proportions are not known, a uniform alpha vector of length equal to the target mixture size is recommended, but the exact value of alpha is still determined by a desired degree of sparsity. For sampling mixtures where a priori proportions are known, the process is very similar, but the alpha vector is scaled by the known proportions. In either scenario, one may need to utilize multiple instantiations of a Dirichlet distribution (i.e., the PyRIID Seed Mixer) to achieve an overall desired sparsity. In conclusion, the setting of the Dirichlet $\alpha$ parameter requires some study by users to understand the nature of sparsity needed for their problem, but the authors find this task more preferable than directly dealing with $\alpha$.

This page intentionally left blank.

# REFERENCES

[1] T. Morrow, N. Price, and T. McGuire, “Pyriid,” 4 2021.

[2] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan, “Matching words and pictures,” *The Journal of Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[3] D. Blei, A. Ng, and M. Jordan, “Latent dirichlet allocation,” *Advances in neural information processing systems*, vol. 14, 2001.

[4] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[5] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[6] J. Mairal, F. Bach, and J. Ponce, “Task-driven dictionary learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 791–804, 2011.

[7] M. Aharon, M. Elad, and A. Bruckstein, “K-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[8] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, “Dictionary learning algorithms for sparse representation,” *Neural computation*, vol. 15, no. 2, pp. 349–396, 2003.

[9] J. A. Tropp, “Just relax: Convex programming methods for identifying sparse signals in noise,” *IEEE transactions on information theory*, vol. 52, no. 3, pp. 1030–1051, 2006.

[10] T. T. Cai and L. Wang, “Orthogonal matching pursuit for sparse signal recovery with noise,” *IEEE Transactions on Information theory*, vol. 57, no. 7, pp. 4680–4688, 2011.

[11] R. Gribonval and K. Schnass, “Dictionary identification — sparse matrix-factorization via $\ell^1$-minimization,” *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3523–3539, 2010.

[12] A. Martins and R. Astudillo, “From softmax to sparsemax: A sparse model of attention and multi-label classification,” in *International conference on machine learning*, pp. 1614–1623, PMLR, 2016.

[13] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, “Sparse convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 806–814, 2015.

[14] X. Sun, N. M. Nasrabadi, and T. D. Tran, “Supervised deep sparse coding networks,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 346–350, IEEE, 2018.

[15] E. Elsen, M. Dukhan, T. Gale, and K. Simonyan, “Fast sparse convnets,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14629–14638, 2020.

[16] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[17] R. Rubinstein, M. Zibulevsky, and M. Elad, “Double sparsity: Learning sparse dictionaries for sparse signal approximation,” *IEEE Transactions on signal processing*, vol. 58, no. 3, pp. 1553–1564, 2009.

[18] A. Y. Yang, Z. Zhou, A. G. Balasubramanian, S. S. Sastry, and Y. Ma, “Fast $\ell^1$-minimization algorithms for robust face recognition,” *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3234–3246, 2013.

[19] Y. Chen, N. M. Nasrabadi, and T. D. Tran, “Hyperspectral image classification using dictionary-based sparse representation,” *IEEE transactions on geoscience and remote sensing*, vol. 49, no. 10, pp. 3973–3985, 2011.

[20] N. Hurley and S. Rickard, “Comparing measures of sparsity,” *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4723–4741, 2009.

[21] O. Taheri and S. A. Vorobyov, “Sparse channel estimation with l p-norm and reweighted l 1-norm penalized least mean squares,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2864–2867, IEEE, 2011.

[22] M. Kloft, U. Brefeld, P. Laskov, K.-R. Müller, A. Zien, and S. Sonnenburg, “Efficient and accurate lp-norm multiple kernel learning,” *Advances in neural information processing systems*, vol. 22, 2009.

[23] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 58, no. 1, pp. 267–288, 1996.

[24] H. Zou, “The adaptive lasso and its oracle properties,” *Journal of the American statistical association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[25] H. Dalton, “The measurement of the inequality of incomes,” *The Economic Journal*, vol. 30, no. 119, pp. 348–361, 1920.

[26] S. Rickard and M. Fallon, “The gini index of speech,” in *Proceedings of the 38th Conference on Information Science and Systems (CISS’04)*, 2004.

[27] D. Zonoobi, A. A. Kassim, and Y. V. Venkatesh, “Gini index as sparsity measure for signal reconstruction from compressive samples,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 927–932, 2011.

[28] S. Tangirala, "Evaluating the impact of gini index and information gain on classification using decision tree classifier algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 612–619, 2020.

[29] T. Daniya, M. Geetha, and K. S. Kumar, "Classification and regression trees with gini index," *Advances in Mathematics: Scientific Journal*, vol. 9, no. 10, pp. 8237–8247, 2020.

[30] D. J. Mitchell, L. Harding, G. G. Thoreson, and S. M. Horne, "Gadras detector response function," tech. rep., Sandia National Laboratories, Albuquerque, NM (United States), 2014.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[32] D. Chotikapanich and W. E. Griffiths, "Estimating lorenz curves using a dirichlet distribution," *Journal of Business & Economic Statistics*, vol. 20, no. 2, pp. 290–295, 2002.

[33] H. Hasegawa and H. Kozumi, "Estimation of lorenz curves: a bayesian nonparametric approach," *Journal of Econometrics*, vol. 115, no. 2, pp. 277–291, 2003.

[34] I. M. Abdalla and M. Y. Hassan, "Maximum likelihood estimation of lorenz curves using alternative parametric model," *Advances in Methodology and Statistics*, vol. 1, no. 1, pp. 109–118, 2004.

[35] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[36] A. Van Omen, "A semi-supervised model for multi-label radioisotope classification and out-of-distribution detection," Master's thesis, University of Michigan (Ann Arbor), 2023.

[37] E. C. Finn, L. A. Metz, R. F. Payne, J. I. Friese, L. R. Greenwood, J. D. Kephart, B. D. Pierson, and T. A. Ellis, "Methods to collect, compile, and analyze observed short-lived fission product gamma data," tech. rep., Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2011.

This page intentionally left blank.

## DISTRIBUTION

### Email—Internal

| Name | Org. | Sandia Email Address |
|---|---|---|
| Technical Library | 1911 | sanddocs@sandia.gov |

### Email—External

| Name | Company Email Address | Company Name |
|---|---|---|
| Hank Zhu | hank.zhu@doe.nnsa.gov | DOE NNSA |
| Jake Zappala | jake.zappala@doe.nnsa.gov | DOE NNSA |

This page intentionally left blank.

This page intentionally left blank.

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.