S1 Statistical Methods Report

Steven Ma(ym432)

December 2024

Word Count: 2906

1 Introduction

This report explores two statistical methods for parameter estimation in a two-dimensional model: Maximum Likelihood Estimation (MLE) and sWeights. The model involves a signal and background component defined over the variables X and Y. The study applies parametric bootstrapping and maximum likelihood fitting techniques to estimate the model parameters. Data generation methods are compared, including the accept-reject sampling method and the use of the percent point function (ppf). The primary aim is to evaluate the bias, uncertainty and computational efficiency of each estimation method, ultimately determining which approach is most appropriate for different scenarios.

2 Signal and Background Model Overview

This model aims to describe a physical system consisting of a signal and background components, both defined over two variables, X and Y, with the following ranges:

$$X \in [0, 5], \quad Y \in [0, 10].$$

2.1 Signal Distribution in X: $g_s(X)$

The signal distribution in X, denoted $g_s(X)$, follows a Crystal Ball distribution, which consists of a Gaussian core and a power-law tail. It is given by:

$$g_s(X) = \begin{cases} N \cdot e^{-Z^2/2}, & \text{if } Z > -\beta, \\ N \cdot \left(\frac{m}{\beta}\right)^m e^{-\beta^2/2} \left(\frac{m}{\beta} - \beta - Z\right)^{-m}, & \text{if } Z \leq -\beta, \end{cases} \tag{1}$$

where $Z = \frac{X-\mu}{\sigma}$ is the standardized variable.

The normalization factor N ensures that the distribution integrates to 1 over the the real domain. In Section ?, it will be shown that the inverse of the normalization constant, N^{-1}, takes the form:

$$N^{-1} = \sigma \left[\frac{m}{\beta(m-1)} e^{-\beta^2/2} + \sqrt{2\pi}\Phi(\beta) \right], \tag{2}$$

where $\Phi(\beta)$ is the cumulative distribution function of the standard normal distribution.

The parameters of the Crystal Ball distribution are as follows:

- $\mu = 3$: The mean of the Gaussian core, which determines the location of the peak of the distribution.
- $\sigma = 0.3$: The standard deviation of the Gaussian core, which controls the width of the distribution.
- $\beta = 1$: A parameter that governs the transition between the Gaussian core and the power-law tail.
- $m = 1.4$: The exponent that defines the slope of the power-law tail.

2.2 Signal Distribution in Y: $h_s(Y)$

The signal distribution in Y, denoted $h_s(Y)$, follows an exponential decay:

$$h_s(Y) = \lambda e^{-\lambda Y}, \tag{3}$$

where $\lambda = 0.3$ is the decay constant.

If the distribution is truncated over the range $[Y_{\text{min}}, Y_{\text{max}}]$, the truncated exponential distribution is given by:

$$h_s(Y) = \begin{cases} \frac{\lambda e^{-\lambda Y}}{1-e^{-\lambda(Y_{\text{max}}-Y_{\text{min}})}}, & \text{if } Y_{\text{min}} \leq Y \leq Y_{\text{max}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The denominator $1 - e^{-\lambda(Y_{\text{max}}-Y_{\text{min}})}$ ensures proper normalization of the distribution over the interval $[Y_{\text{min}}, Y_{\text{max}}]$.

2.3 Background Distribution in X: $g_b(X)$

The background distribution in X, denoted $g_b(X)$, is modeled as a truncated uniform distribution over the range $[X_{\text{min}}, X_{\text{max}}]$. It is given by:

$$g_b(X) = \begin{cases} \frac{1}{X_{\text{max}}-X_{\text{min}}}, & \text{if } X_{\text{min}} \leq X \leq X_{\text{max}}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

2.4 Background Distribution in Y: $h_b(Y)$

The background distribution in Y, $h_b(Y)$, is initially a Gaussian normal distribution centered at $\mu_b = 5$ with standard deviation $\sigma_b = 1$:

$$h_b(Y) = \phi_b(\frac{Y-\mu_b}{\sigma_b}) = \frac{1}{\sqrt{2\pi\sigma_b^2}} e^{-\frac{(Y-\mu_b)^2}{2\sigma_b^2}}. \tag{6}$$

Where, $\mu_b = 0$ and $\sigma_b = 2.5$.

The distribution is then truncated over the range $[Y_{\text{min}}, Y_{\text{max}}]$. The truncated form is:

$$h_b(Y) = \begin{cases} \frac{\phi_b(\frac{Y-\mu_b}{\sigma_b})}{\Phi\left(\frac{Y_{\text{max}}-\mu_b}{\sigma_b}\right)-\Phi\left(\frac{Y_{\text{min}}-\mu_b}{\sigma_b}\right)}, & \text{if } Y_{\text{min}} \leq Y \leq Y_{\text{max}}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Where $\Phi(x)$ is the cumulative distribution function (CDF) of the standard normal distribution:

$$\Phi(Y) = \int_{-\infty}^{Y} \phi(t)\, dt, \tag{8}$$

and $\phi(Y) = \frac{1}{\sqrt{2\pi}} e^{-Y^2/2}$ is the standard normal probability density function (PDF).

2.5 Total Distribution

The total distribution is a weighted sum of the signal and background distributions, given by:

$$f(X,Y) = f_s \cdot s(X,Y) + (1 - f_s) \cdot b(X,Y), \tag{9}$$

where $f_s = 0.6$ is the signal fraction and $1 - f_s = 0.4$ is the background fraction.

Since X and Y are independent for both the signal and background, the joint distributions for the signal and background are:

$$s(X,Y) = g_s(X) \cdot h_s(Y), \tag{10}$$

$$b(X,Y) = g_b(X) \cdot h_b(Y). \tag{11}$$

3 (a) Normlization Constant for Crystall Ball Distribution

To ensure the Crystal Ball distribution $g_s(X;\mu,\sigma,\beta,m)$ is normalized, the condition is given by:

$$N\left[\underbrace{\int_{-\infty}^{-\beta}\left(\frac{m}{\beta}\right)^m e^{-\beta^2/2}\left(\frac{m}{\beta}-\beta-Z\right)^{-m}dZ}_{\text{Integral (1)}}+\underbrace{\int_{-\beta}^{\infty}e^{-Z^2/2}\,dZ}_{\text{Integral (2)}}\right]=1, \tag{12}$$

Where need to solve for N.

For **Integral (1)**, we start with $Z=\frac{X-\mu}{\sigma}$. As $Z\to-\infty$, $X\to-\infty$. When $Z=-\beta$, we have: $X=\mu-\beta\sigma$. Thus, Integral (1) becomes:

$$\int_{-\infty}^{-\beta}\left(\frac{m}{\beta}\right)^m e^{-\beta^2/2}\left(\frac{m}{\beta}-\beta-Z\right)^{-m}dZ.$$

To simplify, let:

$$u=\frac{m}{\beta}-\beta-\frac{X-\mu}{\sigma},\quad du=-\frac{dX}{\sigma}.$$

As $X\to-\infty$, $u\to\infty$. When $X=\mu-\beta\sigma$, then $Z=-\beta$ and $u=\frac{m}{\beta}$.
Rewriting the integral:

$$\int_{-\infty}^{-\beta}\left(\frac{m}{\beta}\right)^m e^{-\beta^2/2}\left(\frac{m}{\beta}-\beta-Z\right)^{-m}dZ=\left(\frac{m}{\beta}\right)^m e^{-\beta^2/2}\int_{\infty}^{\frac{m}{\beta}}u^{-m}\,du\,\sigma.$$

Now evaluate the integral of u^{-m}:

$$\int_{\infty}^{\frac{m}{\beta}}u^{-m}du=\left[\frac{u^{1-m}}{1-m}\right]_{\infty}^{\frac{m}{\beta}}=\frac{1}{m-1}\left(\frac{m}{\beta}\right)^{1-m}.$$

Thus, Integral (1) becomes:

$$\left(\frac{m}{\beta}\right)^m e^{-\beta^2/2}\frac{1}{m-1}\left(\frac{m}{\beta}\right)^{1-m}\sigma.$$

For **Integral (2)**, we also start with $Z=\frac{X-\mu}{\sigma}$. When $Z=-\beta$, $X=\mu-\beta\sigma$. As $Z\to\infty$, $X\to\infty$. From $dZ=\frac{dX}{\sigma}$, we rewrite the integral as:

$$\int_{-\beta}^{\infty}e^{-Z^2/2}dZ=\int_{\mu-\beta\sigma}^{\infty}e^{-\frac{(X-\mu)^2}{2\sigma^2}}\frac{dX}{\sigma}.$$

Let $u=\frac{X-\mu}{\sigma}$, so $du=\frac{dX}{\sigma}$. The limits transform as follows: - When $X=\mu-\beta\sigma$, $u=-\beta$. - When $X\to\infty$, $u\to\infty$.

Substituting into the integral:

$$\int_{\mu-\beta\sigma}^{\infty}e^{-\frac{(X-\mu)^2}{2\sigma^2}}\frac{dX}{\sigma}=\int_{-\beta}^{\infty}e^{-u^2/2}du.$$

We recognize this as the tail of the standard normal cumulative distribution function (CDF). Thus:

$$\int_{-\beta}^{\infty}e^{-Z^2/2}dZ=\sqrt{2\pi}\sigma\left[1-\Phi(-\beta)\right]=\sqrt{2\pi}\sigma\Phi(\beta),$$

Due to the symmetry of the standard normal distribution, the CDF satisfies $\Phi(\beta)=1-\Phi(-\beta)$.

From the normalization condition, we have:

$$N\left[\left(\frac{m}{\beta}\right)^m e^{-\beta^2/2}\frac{1}{m-1}\sigma+\sqrt{2\pi}\sigma\Phi(\beta)\right]=1.$$

Rearranging for N, we obtain:

$$N^{-1}=\sigma\left[\frac{m}{\beta(m-1)}e^{-\beta^2/2}+\sqrt{2\pi}\Phi(\beta)\right].$$

4 (b) Defining and Validating PDFs with Code

The signal and background PDFs described in Section 2 are implemented using the `scipy.stats` [1] library.

For example, the signal distribution in Y, $h_s(Y)$, is implemented as:

```python
from scipy.stats import expon

def smodel_y(y, lambda_s):
    '''Signal distribution on Y: Truncated Exponential distribution'''
    expon_dist = expon(scale=1/lambda_s)
    norm_const = expon_dist.cdf(Y_max) - expon_dist.cdf(Y_min)
    return expon_dist.pdf(y) / norm_const
```

The total model is defined as described in Equation 9.

4.1 Validation of Normalization

The normalization of the PDFs is validated by performing numerical integration using `scipy.integrate.quad` for one-dimensional (1D) distributions and a custom implementation for two-dimensional (2D) distributions. The custom implementation of 2D integration is preferred over `scipy.integrate.dblquad` due to the latter's unintuitive parameter order. The 2D integration is performed by first integrating over X, followed by Y, as shown below:

```python
def integrate_2d(func, x_min, x_max, y_min, y_max, args=()):
    """
    Perform 2D integration of the given function f(x, y).
    Integrate over x and then over y.
    """
    def integrand(y, *args):
        integral_result, _ = quad(func, x_min, x_max, args=(y,) + args)
        return integral_result
    total_integral, _ = quad(integrand, y_min, y_max, args=args)
    return total_integral
```

The integration results are as follows:

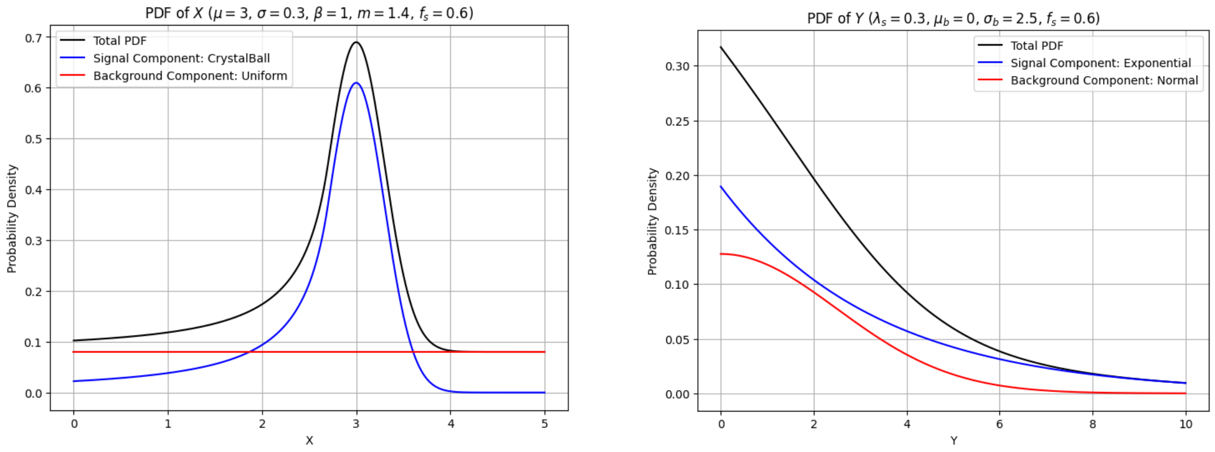
Function	**Integration Range**	**Result**
$g_s(X)$	$X : [0, 5]$	1.0000000376260723
$h_s(Y)$	$Y : [0, 10]$	1.0
$g_b(X)$	$X : [0, 5]$	1.0000000000000002
$h_b(Y)$	$Y : [0, 10]$	1.0
$s(X, Y)$	$X : [0, 5], Y : [0, 10]$	1.0000000376260723
$b(X, Y)$	$X : [0, 5], Y : [0, 10]$	1.0
$f(X, Y)$	$X : [0, 5], Y : [0, 10]$	1.0000000225756431

Table 1: Numerical integration results over specified ranges.

The numerical integration results are very close to 1. These small deviations confirm the correctness of the normalization, with errors attributed to machine precision limitations.

5 (c) Visualization of Probability Distributions

The one-dimensional PDF plots are presented below, X- and Y-axis projections are displayed in Figure 1(a) and 1(b) respectively.



(a) (b)

Figure 1: One-dimensional PDFs. (a) PDF of X; (b) PDF of Y.

The 2D PDF plot as seen in 2 is generated by evaluating the function $f(X, Y)$on a grid created using `numpy.meshgrid`.

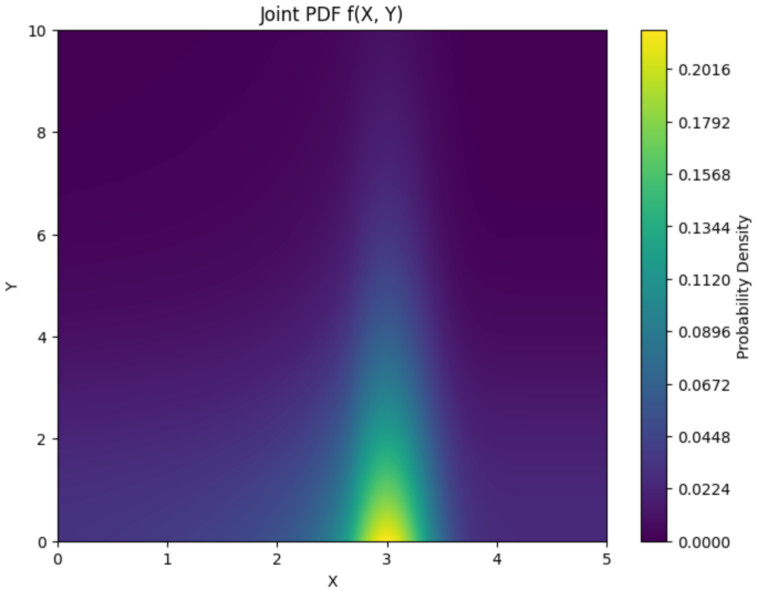


Figure 2: Joint probability density function $f(X, Y)$.

6 (d) Sample Generation and Likelihood Estimation

6.1 Sample Generation: Accept-Reject Method

The signal events are generated using the accept-reject method. The process can be summarized as follows:

- **Candidate Sample Generation:** Generate candidate samples x and y from a uniform distribution within the ranges $[X_{\min}, X_{\max}]$ and $[Y_{\min}, Y_{\max}]$, respectively.
- **Probability Density Evaluation:** For each candidate pair (x, y), evaluate its probability density under the signal distribution $s(x, y)$.
- **Accept-Reject Criterion:**
  1. Generate a random number from uniform distribution$u \sim U(0, 1)$.
  2. Accept the sample (x, y) if $u < s(x, y)$; otherwise, reject the sample and resample.

- **Iteration:** Repeat the above steps until N_{s} valid signal events are generated.

The background events are generated in a similar manner, using the background probability density function $b(X,Y)$. The total dataset is created by combining the signal and background samples. The number of signal and background events is determined by:

$$N_s = N_{\mathrm{tot}} \cdot f_s, \quad N_b = N_{\mathrm{tot}} - N_s,$$

where N_{tot}(or N) is the total number of events, N_s is the number of signal events, and N_b is the number of background events.

In this section, N_{tot} is fixed to 100,000, so:

$$N_s = 60,000, \quad N_b = 40,000.$$

Advantages: The accept-reject method does not require the target distribution's inverse CDF (e.g., Percent-Point Function PPF) to be known or computable. So it can be applied to any probability density function, regardless of its form.

Disadvantages: The efficiency of the accept-reject method depends on the shape of the target distribution. If the probability density is close to zero over most of the sampling region, the acceptance rate becomes low. In this case, generating a dataset of 100,000 samples achieves only a 2% acceptance rate, resulting in longer computation times.

6.2 Sample Generation: PPF

The PPF is the inverse of the CDF. It maps a uniform random variable $u \sim U(0,1)$ onto a value from the target probability distribution, the sampled data described in Equation 9 can be generated as follows:

Signal Events: The X-coordinates for signal events are sampled from $g_s(x)$, and the Y-coordinates from $h_s(y)$. This involves:

1. Computing the CDFs $G_s(x)$ and $H_s(y)$ over the respective ranges $[X_{\mathrm{min}}, X_{\mathrm{max}}]$ and $[Y_{\mathrm{min}}, Y_{\mathrm{max}}]$.
2. Generating uniform random numbers $u_x \in [G_s(X_{\mathrm{min}}), G_s(X_{\mathrm{max}})]$ and $u_y \in [H_s(Y_{\mathrm{min}}), H_s(Y_{\mathrm{max}})]$.
3. Transforming u_x and u_y into X_{signal} and Y_{signal} using the PPFs $G_s^{-1}(u_x)$ and $H_s^{-1}(u_y)$.

Background Events: The X-coordinates are directly sampled from the uniform distribution $U(X_{\mathrm{min}}, X_{\mathrm{max}})$. The Y-coordinates are sampled from $h_b(y)$ by:

1. Computing the CDF $H_b(y)$ for $[Y_{\mathrm{min}}, Y_{\mathrm{max}}]$.
2. Generating uniform random numbers $u_b \in [H_b(Y_{\mathrm{min}}), H_b(Y_{\mathrm{max}})]$.
3. Transforming u_b into $Y_{\mathrm{background}}$ using the PPF $H_b^{-1}(u_b)$.

The final dataset is formed by combining signal and background events.

Advantages: The PPF enables direct sampling from the target probability distribution, eliminating the need for iterative rejection or candidate generation, resulting in computational efficiency.

Disadvantages: The PPF method requires an analytically defined or numerically computable PPF, which may not exist for certain complex distributions.

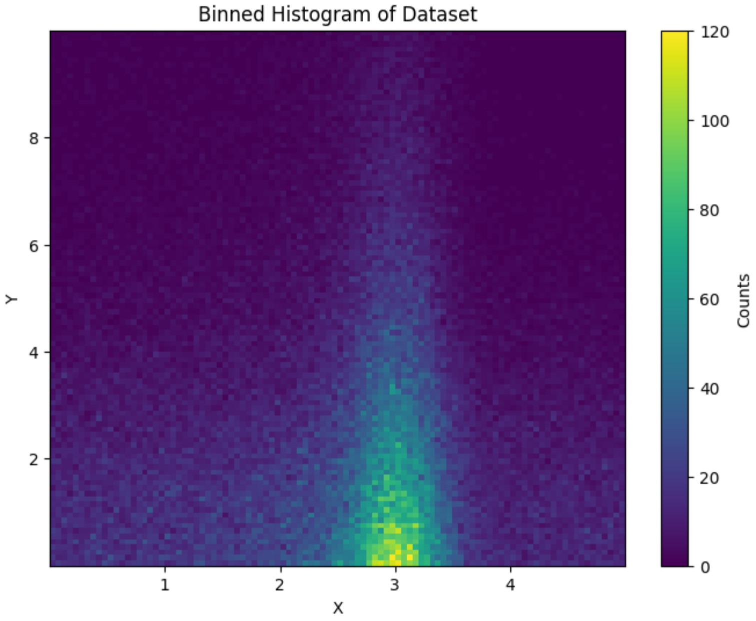


Figure 3: Binned histogram of the generated dataset.

The histogram shown in Figure 3 matches well with the joint probability density function (PDF) presented in Figure 2. Both plots exhibit similar structural patterns, particularly the concentration of events around $X = 3$ and $Y = 0$, which is consistent with the underlying signal and background distributions.

Furthermore, the histograms generated using the PPF and the Accept-Reject methods are indistinguishable by eye. Performing fits on both datasets yields reasonable parameter estimates. This agreement confirms the correctness and reliability of the generated samples.

6.3 Extended Maximum Likelihood Estimates

The extended likelihood function used for parameter estimation is given by:

$$\mathcal{L}(\vec{\theta}; \text{data}) = \frac{e^{-\mu_{\text{total}}} \mu_{\text{total}}^N}{N!} \prod_{i=1}^{N} \left[f_s \cdot s(X_i, Y_i; \mu, \sigma, \beta, m, \lambda) + (1 - f_s) \cdot b(X_i, Y_i; \mu_b, \sigma_b) \right], \tag{13}$$

where:

- $\mathcal{L}$: The likelihood function.
- μ_{total}: The total expected number of events (signal and background combined).
- N: The observed number of events.
- $\vec{\theta}$: The full set of 9 parameters to estimate, including $\mu, \sigma, \beta, m, f_s, \lambda, \mu_b, \sigma_b, \mu_{\text{total}}$.

In this section, the number of observation N is not generated from Poisson distribution, but it's still meaningful to estimate μ_{tot} for comparision purposes.

Explanation: The likelihood function combines two components:

1. **Poisson Term:** The factor $\frac{e^{-\mu_{\text{total}}} \mu_{\text{total}}^N}{N!}$ accounts for the probability of observing N events given the total expected number of events μ_{total}.
2. **Per-Event Contribution:** The product term represents the probability of observing each individual event.

In practice, Equation 13 is very difficult to maximize directly. Since ln is monotonic, we take the negative natural logarithm of the likelihood function to simplify the optimization process. The negative log-likelihood becomes:

$$-\ln \mathcal{L} = \mu_{\text{total}} - N \ln \mu_{\text{total}} + \ln N! + \sum_{i=1}^{N} \ln \left(f_s \cdot s(X_i, Y_i; \mu, \sigma, \beta, m, \lambda) + (1 - f_s) \cdot b(X_i, Y_i; \mu_b, \sigma_b) \right). \tag{14}$$

`iminuit.cost.ExtendedUnbinnedNLL` is implemented to minimize this function[2]. A typical fit results for the parameters are as follows:

$$\begin{aligned} f_s &= 0.6018 \pm 0.0035, & \mu &= 3.0021 \pm 0.0026, & \sigma &= 0.3002 \pm 0.0024, \\ \beta &= 1.0133 \pm 0.0222, & m &= 1.3305 \pm 0.0583, & \lambda_s &= 0.2982 \pm 0.0020, \\ \mu_b &= 0.0000 \pm 0.0669, & \sigma_b &= 2.5017 \pm 0.0190, & \mu_{tot} &= 100000.26 \pm 316.23. \end{aligned}$$

Parameter Correlations

In an ideal scenario, if the parameter estimates follow a normal distribution and their uncertainties are accurately estimated, approximately 68% of true values should fall within the fitted value's ±1σ range[3]. For 500 fits, each using 100,000 samples, the expected number of hits for each parameter is approximately 340.

Parameter	f_s	μ	σ	β	m	λ_s	μ_b	σ_b	μ_{tot}
Counts	373	324	327	293	316	330	453	394	500

Table 2: True value hits within the ±1σ range across 500 MLE fits. Expected hit count: 340 (68%).

For the original, untruncated distribution, the parameters can be considered independent. However, when the distribution is truncated, dependencies between the parameters are introduced.

The parameters μ, σ, β, and m are part of the truncated Crystal Ball function and exhibit varying degrees of correlation (see Table 3). Among them, β and m are highly correlated, while μ and σ show moderate correlation.

- μ (324 hits) and σ (327 hits): Slightly below the expected 340 hits, suggesting that the uncertainties might be slightly underestimated
- β (293 hits) and m (316 hits): Significantly below the expected 340 hits, particularly for β, which only has 293 hits. This indicates either underestimation of uncertainties or systematic biases in their estimates.
- f_s (373 hits): This independent parameter shows weak correlations with other parameters. Its hit count exceeds the expected 340, indicating that the uncertainty is slightly overestimated or the parameter distribution is narrower.
- λ_s (330 hits): This model-specific parameter is close to the expected 340 hits, indicating good agreement with theoretical expectations.
- μ_b (453 hits) and σ_b (394 hits): Both are well above the expected 340 hits, especially μ_b, with 453 hits. This suggests that the uncertainties for these parameters are overestimated, or their fits to the data are exceptionally strong.
- μ_tot (500 hits): Always within the ±1σ range. Its perfect agreement reflects no correlation with other parameters.

The correlation between σ_b and μ_b is highly negative (see Table 3), indicating strong linear dependency. Strong correlations like this can negatively impact the stability and accuracy of Maximum Likelihood Estimation (MLE). However, despite the high correlation, the parameter estimates for μ_b and σ_b appear robust. This issue has been noted, but further analysis is beyond the scope of this work.

	f_{s}	μ	σ	β	m	λ_s	μ_b	σ_b	μ_{tot}
f_{s}	1	0	0.3	0.1	-0.4	0.3	0.3	-0.3	0
μ	0	1	-0.5	-0.5	0.3	0	0	0	0
σ	0.3	-0.5	1	0.4	-0.3	0.1	0.1	-0.1	0
β	0.1	-0.5	0.4	1	-0.9	0	0.1	-0.1	0
m	-0.4	0.3	-0.3	-0.9	1	-0.1	-0.2	0.2	0
λ_s	0.3	0	0.1	0	-0.1	1	0.1	0	0
μ_b	0.3	0	0.1	0.1	-0.2	0.1	1	-1	0
σ_b	-0.3	0	-0.1	-0.1	0.2	0	-1	1	0
μ_{tot}	0	0	0	0	0	0	0	0	1

Table 3: A typical correlation matrix of fitted parameters.

6.4 Time Benchmark

From Table 4, it is evident that generating 100,000 samples using the PPF method is approximately 70 times faster than the Accept-Reject method. In this task, the PPFs for all models are available, and both methods produce consistent datasets. Therefore, for this project, the PPF method is more suitable for data generation. Consequently, all subsequent analyses use datasets generated via the PPF method.

The fit time ranges from a minimum of 14 seconds to a maximum of 45 seconds, which can be attributed to variations in computational resource usage and system load. The fit time presented in Table 4 reflects a reasonably optimal case.

Task	**Time (seconds)**	**Relative Time**
`np.random.normal`	0.001345	1.00x
PPF Method (avg. over 100)	0.024204	18.00x
Accept-Reject Method (, avg. over 100)	1.685230	1253.39x
MLE Fit (avg. over 100)	14.834084	11032.85x

Table 4: Computation Times for Generating 100,000 Samples and MLE Fit.

7 (e) Parametric Booststrapping for Estimating λ

The goal of this simulation study is to investigate the bias and uncertainty of the parameter λ, as a function of sample size N. Parametric bootstrapping is employed using an ensemble of 250 simulations for each sample size. Sample sizes of 200, 500, 1000, $2,500$, $5,000$, $10,000$ and $30,000$ are considered, with Poisson variations applied to the sample size to simulate realistic fluctuations.The datasets are generated from the true distribution.

For each sample size:

1. Generate synthetic datasets using the PPF with the specified sample size.
2. Perform MLE to fit the parameters of the model.
3. Calculate the mean, standard deviation, and bias of the estimated parameters over 250 simulations.

It can be concluded from Figure 4 that:

- **Bias:** As the sample size increases, the bias of all parameters decreases and approaches zero, which indicates that the parameter estimates become more accurate with larger sample sizes. This trend is consistent across all parameters
- **Uncertainty (Standard Deviation):** Standard deviation can be estimated as:

$$\hat{\sigma} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{15}$$

  The uncertainty of the parameter estimates decreases as the inverse square root of the sample size, $\hat{\sigma} \propto \frac{1}{\sqrt{N}}$, where N is the sample size. This is true for all parameters.

The results in Table 5 demonstrate that the bias of λ decreases steadily with increasing sample size, approaching zero for larger samples. Similarly, the standard deviation decreases consistently.

Sample Size	**Bias**	**Standard Deviation**
200	0.0067	0.0466
500	0.0006	0.0310
1000	0.0008	0.0215
2500	0.0010	0.0140
5000	0.0005	0.0089
10000	0.0000	0.0065
30000	0.0001	0.0039

Table 5: Bias and Standard Deviation of λ at Different Sample Sizes (pure MLE, 4 s.f.)

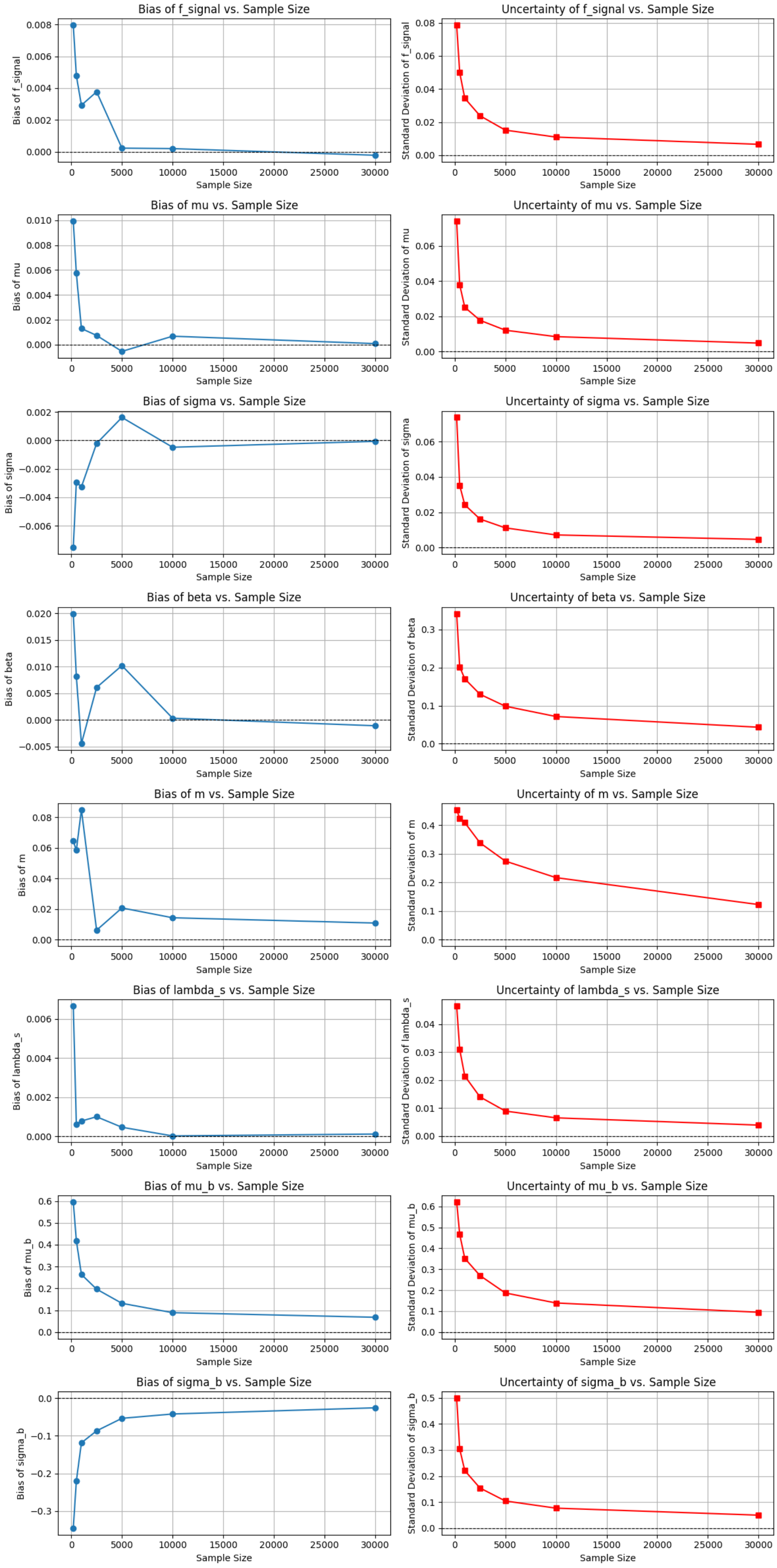


Figure 4: Bias and standard deviation of all parameters as a function of sample size.

8 (f) sWeights for Estimating λ

8.1 Theoretical Foundations

The sWeights technique is used for projecting the signal and background components in a multi-dimensional space. Consider only the x componet of Equation 9:

$$g(X) = f_s \cdot g_{\mathrm{s}}(X) + (1 - f_s) \cdot g_{\mathrm{b}}(X) \tag{16}$$

The key method is the construction of the weight matrix W, which describes the overlap between signal and background components in the feature space X:

$$W = \begin{pmatrix} W_{ss} & W_{sb} \\ W_{bs} & W_{bb} \end{pmatrix}$$

The elements of W are defined as integrals over the product of the component distributions [4]:

$$W_{ss} = \int \frac{g_s^2(X)}{g(x)} dX, \quad W_{sb} = W_{bs} = \int \frac{g_x(X) \cdot g_b(X)}{g(X)} dX, \quad W_{bb} = \int \frac{g_b^2(x)}{g(X)} dX$$

The inverse matrix $\alpha = W^{-1}$ is then computed:

$$\alpha = \frac{1}{\det(W)} \begin{pmatrix} W_{bb} & -W_{sb} \\ -W_{bs} & W_{ss} \end{pmatrix} = \frac{1}{\det(W)} \begin{pmatrix} \alpha_{ss} & \alpha_{sb} \\ \alpha_{bs} & \alpha_{bb} \end{pmatrix}$$

Each event's sWeights are calculated using:

$$w_i^s = \alpha_{ss} \cdot \frac{g_{\mathrm{s}}(X_i)}{g(X_i)} + \alpha_{sb} \cdot \frac{g_{\mathrm{b}}(X_i)}{g(X_i)}, \quad w_i^b = \alpha_{bs} \cdot \frac{g_{\mathrm{s}}(X_i)}{g(X_i)} + \alpha_{bb} \cdot \frac{g_{\mathrm{b}}(X_i)}{g(X_i)} \tag{17}$$

These sWeights, w_i^s and w_i^b, quantify the contribution of each event to the signal and background components, respectively. Notably, these weights can take on negative values or exceed one. Importantly, for each event, the sum of w_i^s and w_i^b is equal to one.

Construction of Histogram

To construct the weighted histograms in the Y-direction for both signal and background components, the following procedure is employed:

- **Define Y Bins**: The Y variable is divided into M intervals ranging from Y_{min} to Y_{max}.
- **Initialize Histograms**: Two arrays, $B^s(Y)$ and $B^b(Y)$, each of length M, are initialized to zero to store the weighted counts for signal and background respectively.
- **Populate Histograms**: For each event i, identify the bin j such that $Y_i \in [Y_{j-1}, Y_j)$. Update the histograms as follows:

$$B^s(Y_j) = \sum_{i=1}^{N} w_i^s \cdot \delta(Y_i \in [Y_{j-1}, Y_j)), \quad B^b(Y_j) = \sum_{i=1}^{N} w_i^b \cdot \delta(Y_i \in [Y_{j-1}, Y_j)) \tag{18}$$

Here, δ represents the Dirac delta function, and $B(Y_j)$ denotes the weighted frequency in bin j. The weights w_i^s and w_i^b can be negative or exceed one, but they satisfy $w_i^s + w_i^b = 1$ for each event i. These weighted histograms, $B^s(Y_j)$ and $B^b(Y_j)$, provide the projections of the signal and background distributions in the Y-direction, respectively.

Event Counts

Signal event counts N_s and background event counts N_b can be estimated by[5]:

$$N_s = \sum_{i=1}^{N} w_i^s \quad N_b = \sum_{i=1}^{N} w_i^b, \tag{19}$$

8.2 Estimating λ

In this section, the discriminant variable X is used to distinguish between signal and background through its distribution $g(X)$ (Equation. 16), enabling the calculation of sWeights.

The control variable Y is then analyzed using the sWeights to isolate the signal distribution and estimate the decay constant λ.

The procedure is as follows: using X-direction data alone, the extended likelihood function

$$\mathcal{L}_x(\mu, \sigma, \beta, m, \mu_{tot}; \vec{X}) = \frac{e^{-\mu_{\text{total}}}\mu_{\text{total}}^N}{N!}\prod_{i=1}^{N}[f_s \cdot g_{\text{s}}(X_i) + (1-f_s)\cdot g_{\text{b}}(X_i)] \tag{20}$$

is fitted using MLE to obtain the best-fit parameters μ, σ, β, m, and μ_{tot}.

Then, The sWeights w_i^s and w_i^b are then calculated using `sweights.SWeight`[6]. The resulting sWeights in the X-direction are shown in Fig. 5, where the signal sWeights and background sWeights are plotted with their sum, which equals 1.

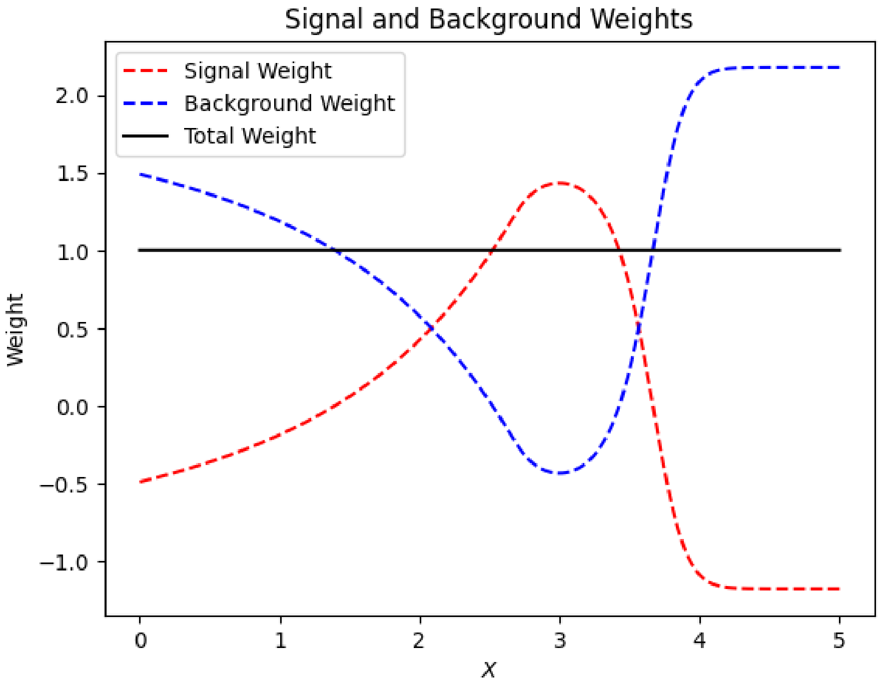


Figure 5: Signal and background weights in X direction.

Projection to the Y-Direction

The signal sWeights w^s and background sWeights w^b are applied to the Y-values of the dataset to construct histogram and isolate the signal contribution, as given by Eq. 18.

The histogram is constructed using $M = 100$ bins across the Y-range. The resulting unnormalized projections of the signal and background distributions are shown in Figure 6. The left panel displays the signal distribution weighted by w^s, and the right panel shows the background distribution weighted by w^b. By visual inspection, the resulting signal and background distributions appear consistent with those shown in Figure 1.

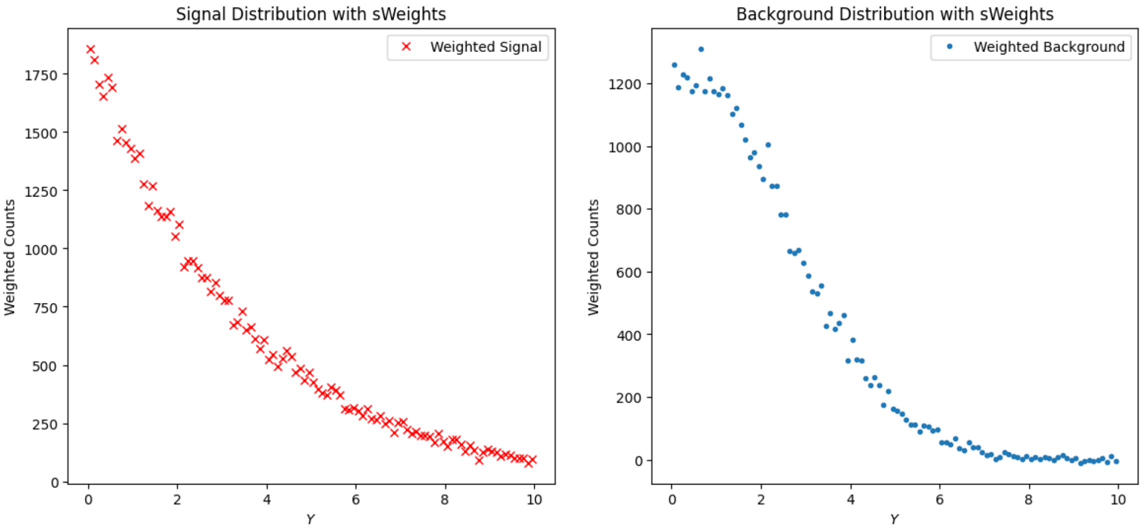


Figure 6: Projections of the signal (left) and background (right) distributions in the Y-direction using sWeights.

Binned MLE fit for λ

The expected count λ_j in the j-th bin is:

$$\lambda_j = N_s \cdot h_s(Y_j) \cdot \Delta Y_j \tag{21}$$

where:

- $N_s = \sum_{j=1}^{M} B^s(Y_j)$ is the total observed count across all bins.
- ΔY_j is the width of the j-th bin.

Assuming the observed signal counts $B^s(Y_j)$ in bin j follow a Poisson distribution with parameter λ_j, the likelihood contribution for the j-th bin is:

$$\mathcal{L}_j = \frac{(\lambda_j)^{B^s(Y_j)} e^{-\lambda_j}}{B^s(Y_j)!} \tag{22}$$

The total likelihood across all bins is the product of the individual bin contributions:

$$\mathcal{L}_{bin}(\lambda; B^s(Y)) = \prod_{j=1}^{M} \frac{(\lambda_j)^{B^s(Y_j)} e^{-\lambda_j}}{B^s(Y_j)!} \tag{23}$$

Substituting the expression for λ_j from Equation (21) and taking the negative natural logarithm of the likelihood function, we obtain the negative log-likelihood:

$$-\ln \mathcal{L}_{bin}(\lambda; B^s(Y)) = \sum_{j=1}^{M} \left[B^s(Y_j) \ln \left(N_s \cdot h_s(\lambda; Y_j) \cdot \Delta Y_j \right) - N_s \cdot h_s(\lambda; Y_j) \cdot \Delta Y_j \right] \tag{24}$$

Minimizing Equation 24, the estimates of λ can be obtained. Based on Equation 19, the total number of events N can also be estimated.

To compare the methods: sWeights and the pure MLE method described in Section 7, 250 simulations were conducted for each sample size, with the sample sizes considered being $\{200, 500, 1000, 2500, 5000, 10000, 30000\}$. The average biases and standard deviations of λ are summarized in Table 6. Note that this analysis employs a binned fit, the method in Section 7 uses an unbinned fit due to implementation constraints.

Table 6: Bias and Standard Deviation of λ for Different Sample Sizes (sWeights).

Sample Size	Bias	Standard Deviation
200	0.0020	0.0487
500	-0.0009	0.0315
1000	-0.0000	0.0206
2500	0.0012	0.0144
5000	-0.0006	0.0096
10000	0.0004	0.0069
30000	-0.0003	0.0041

Figure 7 and Figure 8 compare the estimated bias and standard deviation of λ, as well as the relative bias and relative standard deviation of the total event count N.

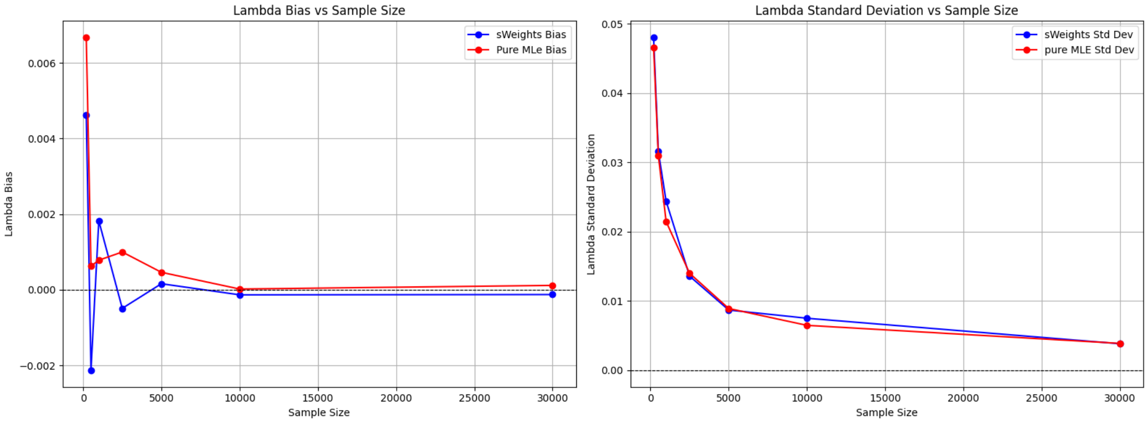


Figure 7: Bias and Standard Deviation for λ between the sWeights method and pure MLE method.

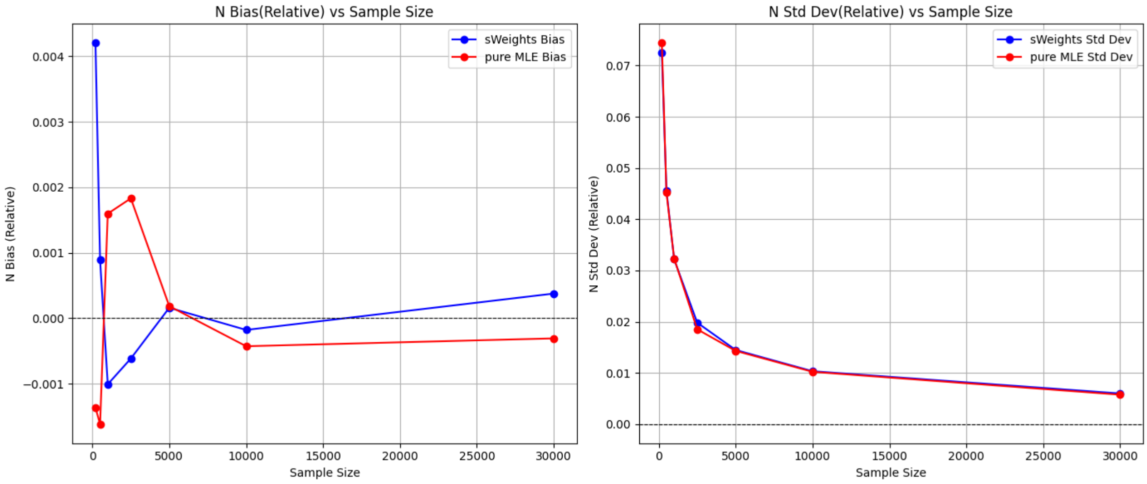


Figure 8: Relative bias (Bias/N) and relative standard deviation (Std Dev/N) of the total event count N for pure MLE and sWeights methods.

Both methods exhibit:

- Bias approaching zero as the sample size increases, becoming very small by $N = 30{,}000$.
- Standard deviation scaling as $\sim 1/\sqrt{N}$.

Empirical results demonstrate that both methods exhibit asymptotically unbiased behavior.

9 (f) Estimating λ Comparison: Pure MLE vs. sWeights with MLE

9.1 Computational Time

In this work, A direct comparison of computational time between the two methods is inappropriate, as pure MLE employs an unbinned fit while sWeights uses a binned fit. For a fair comparison, both methods should adopt a binned fit. However, due to time constraints, this was not conducted.

Theoretical analysis suggests that binned fits scale roughly as $O(K^2)$, where K is the number of parameters to fit [7]. In pure MLE, all 9 parameters are fitted simultaneously. In contrast, the sWeights method first fits 5 parameters (in the X-direction) and subsequently fits a single parameter λ using the separated Y-signal data.

Assuming identical binned fits and recognising fitting is the most time-consuming step, the sWeights method is estimated to be:

$$\frac{9^2}{5^2+1^2} \approx 3.11$$

times faster. Time constraints have prevented the full experimental validation of this efficiency gain.

9.2 Applicability

sWeights:

- Enables the separation of contributions from different event sources, yielding statistically pure distributions[5].
- The projected distribution in the control variable is an unbiased estimator of the true distribution[5].
- Does not require prior knowledge of the control variable's distribution; it infers the distribution using information from the discriminant variable (X in this work).
- Assumes the control variable and the discriminant variable are uncorrelated; this assumption can be tested using Monte Carlo simulations[4].
- Inherently generates binned data, making it computationally efficient and well-suited for large datasets and signal-background separation.

Pure MLE:

- Applicable to both binned and unbinned data, preserving all statistical information in the dataset.
- Allows the simultaneous estimation of multiple parameters, offering flexibility for general statistical analyses.
- Particularly suitable for small datasets where high precision is essential.
- Computational complexity increases significantly with the number of parameters.
- Reliability depends heavily on the accuracy of the assumed PDFs, which requires prior knowledge.

10 Conclusion

Both the pure MLE and sWeights methods provide accurate and consistent parameter estimates for λ, with bias decreasing and uncertainty improving as the sample size increases. The sWeights method efficiently separates signal and background contributions, enabling a focused fit on the signal parameter λ. This modularity reduces the dimensionality of the fit and can offer significant computational advantages, particularly for large datasets. In contrast, pure MLE remains valuable for small datasets where precision is critical and offers a more general framework applicable to a wide range of statistical models.

References

[1] Virtanen, P., Gommers, R., Oliphant, T. E., et al., *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods, 17, pp. 261–272, 2020.

[2] James, F. and Roos, M., *Minuit - a system for function minimization and analysis of the parameter errors and correlations*, Computer Physics Communications, 10(6), pp. 343-367, 1975.

[3] Casella, G. and Berger, R. L., *Statistical Inference*, 2nd ed., Chapter 9: Interval Estimation, Duxbury, 2002.

[4] Pivk, M., *sPlot: A Quick Introduction*, Contributed to PHYSTAT 05 - Oxford, 12th-15th September, arXiv:physics/0602023 [physics.data-an], 2006.

[5] Pivk, M. and Le Diberder, F. R., *sPlot: A statistical tool to unfold data distributions*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 555(1–2), pp. 356-369, 2005.

[6] Dembinski, H., Kenzie, M., Langenbruch, C. and Schmelling, M., *Custom Orthogonal Weight functions (COWs) for event classification*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 1040, 167270, 2022.

[7] Scikit-HEP, *iminuit: Basic Usage Tutorial*, Available at: `https://scikit-hep.org/iminuit/notebooks/basic.html`.

Appendix

Based on the original text and instructions provided by the author, texts in this report was generated by ChatGPT 4o. The tool also assisted in formatting and refining the LaTeX code.