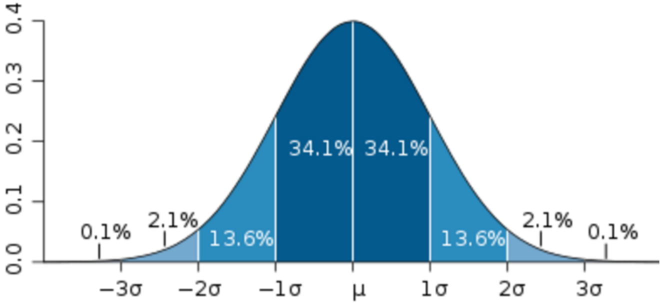
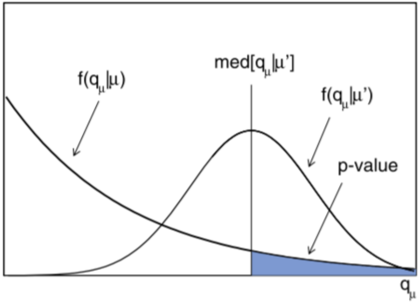




PH2282 part 2: Likelihood and Point Estimation

Applied Multi-Messenger Astronomy 2:
Statistical and Machine Learning Methods in Particle and Astrophysics

Matteo Agostini

TUM - summer term 2019

Tasks of statistical inference

Task name	Task description	Some Frequentist tools	Some Bayesian tools
Point Estimation	what is the best estimate for a parameter of the model?	Maximum likelihood estimator	Median or mode of posterior distribution
Hypothesis Testing	which (model) hypothesis can be accepted or rejected given the data?	likelihood ratios	Bayes factors
Interval Estimation	which range of values is plausible for a given parameter of the model?	inverse hypothesis test	intervals of the posterior distribution
Goodness of Fit	are my data compatible with a model?	chi-square test, likelihood ratio test	posterior predictive p-values

Ingredients of a Frequentist analysis

1) Concept of probability
2) Statistical model and data set
   - Random variables
   - Parameters of the model
   - Probability distribution functions (PDF's)
3) Likelihood function $\mathcal{L}$
4) Estimators [Point Estimation]
5) Test statistics [Hypothesis Testing]
   - power and size of a test
   - Likelihood ratios
6) Confidence interval [Interval Estimation]
   - coverage
   - inverting an hypothesis test

Summary of key concepts

- A **random phenomenon** is a phenomenon whose outcome is unpredictable
- The **Frequentist probability** is the frequency of an outcome for a large (infinite) number of trials
- A **random variable** X is a variable whose possible values x are the outcomes of a random phenomenon. It is a variable in the sense that the frequency of its outcomes depends on the properties of the phenomenon (aka the parameters of a models). It is random in the sense that the outcome of the process is random, ergo unpredictable;
- The **probability distribution** of a random variable $X \sim f(x)$ is a function that associates a probability value for each possible outcome to occur
- A **statistical model** is defined by a set of parameters, a set of basic random variables, and their probability functions
- The probability functions of complex random variables can be constructed as a function of the basic random variables of the model (e.g. the number of counts within an energy bin)

Summary of our statistical model

Parameters of models:
λ_s and λ_b, i.e. the expectation for the total numbers of signal and background events

Elementary random variables:
N number of events (signal and background); X energy of an event

Probability distribution function for a single event energy:

$$X \sim f_X(x; \mu, \sigma, \lambda_s, \lambda_b) = \frac{1}{\lambda_s + \lambda_b}\left(\lambda_s \cdot f_X^s(x; \mu, \sigma) + \lambda_b \cdot f_X^b(x)\right) = \frac{1}{\lambda_s + \lambda_b}\left[\lambda_s \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} + \lambda_b \cdot \frac{1}{20}\right]$$

Probability distribution function for the number of events: background events:

$$N \sim f_N(n; \lambda_s + \lambda_b) = \frac{e^{-(\lambda_s+\lambda_b)}(\lambda_s + \lambda_b)^n}{n!}$$

Questions?

Aside on the parameters of the model

The parameters of a model can be divided into three groups:

- parameters of interest (POI): the parameters on which we want to do statistical inference (e.g. λ_s)
- nuisance parameters: parameters that are not known but whose value is not of interest (e.g. λ_b)
- known/fixed parameters: parameters of the model whose value is known and fixed (e.g. μ and σ)

In our case λ_s is the parameter of interest, λ_b is a nuisance parameter and μ/σ are fixed. Which parameters are nuisance or fixed depends on the theory behind the model. The value of fixed parameters is considered as known with infinite accuracy, while the value of nuisance parameters is typically related to a measurement and is constrained with some uncertainties.

Role of the model in statistical inference

1) **generation of ensemble of pseudo data (toy Monte Carlo):**
   - to understand which data are expected by a model and with which frequency
   - to study the impact of the model parameters on the data
   - to test the analysis pipeline
   - to built probability distributions for complex random variables of the data

2) construction of the joint probability function for the data
   - used as tool for estimating the model parameters, confidence intervals and perform hypothesis testing

Random sampling

- Random sampling means to generate values $\{x_1, \ldots, x_N\}$ for a random variable $X \sim f_X(x)$
- generate random values from a uniform distribution $U \sim$uniform(0,1) is relatively easy, the problem is to sample from not uniform distributions
- Sampling from more complicated distributions can be done through direct random sampling:
  - idea: find a function that transforms random samples of U into random samples of X

Random sampling

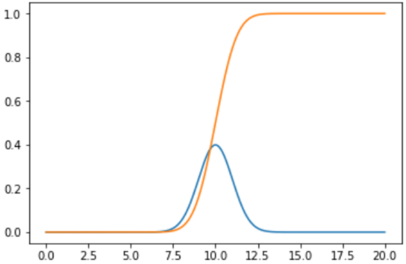
- Random sampling means to generate values $\{x_1, \ldots, x_N\}$ for a random variable $X \sim f_X(x)$
- generate random values from a uniform distribution $U \sim$uniform(0,1) is relatively easy, the problem is to sample from not uniform distributions
- Sampling from more complicated distributions can be done through direct random sampling:
  - idea: find a function that transforms random samples of U into random samples of X
  - tool: the cumulative distribution function (CDF) of f_X:

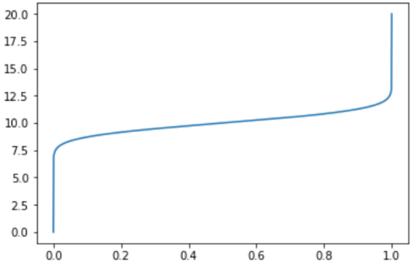
$$F_X(x) = \int_{-\infty}^{x} f_X(t)\, dt$$

$\boxed{\lim_{x\to-\infty} F(x) = 0 \text{ and } \lim_{x\to+\infty} F(x) = 1}$ $\boxed{0 \le F(x) \le 1}$ $\boxed{F(x) \text{ monotonic function}}$

  - The inverse of the CDF is the transformation we need: $F_X : X \to U$ ergo $F_X^{-1} : U \to X$

```python
import numpy as np
import matplotlib.pyplot as pl
from scipy import stats
from scipy.interpolate import interp1d

# define grid of points to describe the pdf
x = np.linspace(0, 20, 10000)

# compute pdf
pdf = stats.norm.pdf(x, loc=10, scale=1)

# compute cumulative distribution
cdf = np.cumsum(pdf); cdf /= max(cdf)

# compute inverse of the CDF
inverse_cdf = interp1d(cdf, x, \
                       bounds_error=False, \
                       fill_value = 0)

u = np.linspace(0, 1, 1000)

pl.plot(x, pdf, x, cdf); pl.show()
pl.plot(u,inverse_cdf(u)); pl.show()
```





Random sampling for our reference model

Steps:

1) Sample a random number of events from a Poisson distribution with expectation $\lambda_{tot} = \lambda_s + \lambda_b$. The random sampling of standard distributions can be done easily using existing functions e.g:

   ```
   N = stats.poisson.rvs(mu=lamnda_tot)
   ```

2) Sample N random values for the energy distribution using the inverse of the CDF

Role of the model in statistical inference

1) generation of ensemble of pseudo data (toy Monte Carlo):
    - to understand which data are expected by a model and with which frequency
    - to study the impact of the model parameters on the data
    - to test the analysis pipeline
    - to built probability distributions for complex random variables of the data

2) **construction of the joint probability function for the data**
    - **used as tool for estimating the model parameters, confidence intervals and perform hypothesis testing**

Joint probabilities

Up to now we have been considering only probabilities for a single random variable. Now we want to define a joint probability for the overall outcome of a set of random variables

What is the probability of two heads up if we toss two coins?

Joint probabilities

Up to now we have been considering only probabilities for a single random variable. Now we want to define a joint probability for the overall outcome of a set of random variables

What is the probability of two heads up if we toss two coins?

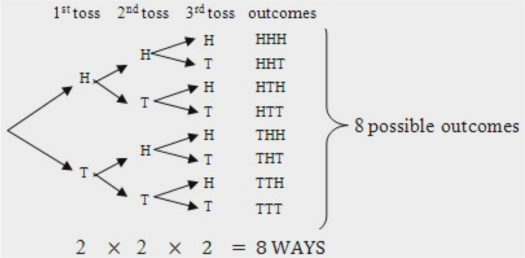
- X is the variable associated to the side of the first coin
- Y is the variable associated to the side of the second coin
- the outcome of a trial is (x, y)
- the possible outcomes are $\{\{H, H\}, \{H, T\}, \{T, H\}, \{T, T\}\}$
- what is $P(\{H, H\})$?

Joint probabilities

What is the probability of three heads up if we toss three coins?

Joint probabilities

What is the probability of three heads up if we toss three coins?



The possible outcomes are

$$\{\{H,H,H\},\{H,H,T\},\{H,T,H\},\{H,T,T\}\},\{\{T,T,T\},\{T,T,H\},\{T,H,T\},\{T,H,H\}\}$$

Joint probability distributions

If two random variables are uncorrelated, the probability of a composite outcome is given by the product of the probability of each outcome:

$$P(X, Y) = P(X) \cdot P(Y)$$

Similarly, the probability distribution function for a composite outcome is

$$f_{X,Y}(x, y) = f_X(x) \cdot f_Y(y)$$

Joint probability distributions

If two random variables are uncorrelated, the probability of a composite outcome is given by the product of the probability of each outcome:

$$P(X, Y) = P(X) \cdot P(Y)$$

Similarly, the probability distribution function for a composite outcome is

$$f_{X,Y}(x, y) = f_X(x) \cdot f_Y(y)$$

In general, if we have a vector of N random variables $\vec{X} = \{X_1, X_2, \ldots, X_N\}$, their joint probability distribution is:

$$f_{\vec{X}}(\vec{x}) = \prod_{i=1}^{N} f_{X_i}(x_i)$$

Joint probability distributions for our reference example

The joint PDF for a vector of N uncorrelated random variables $\vec{X} = \{X_1, X_2, \ldots, X_N\}$ is:

$$f_{\vec{X}}(\vec{x}) = \prod_{i=1}^{N} f_{X_i}(x_i)$$

In our reference example, the probability distribution for the energy of an event is:

$$f_X(x; \mu, \sigma, \lambda_s, \lambda_b) = \frac{1}{\lambda_s + \lambda_b} \left[\lambda_s \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} + \lambda_b \cdot \frac{1}{20} \right]$$

The joint PDF for N events will thus be:

$$f_{\vec{X}}(\vec{x}; \mu, \sigma, \lambda_s, \lambda_b) = \prod_{i=1}^{N} f_X(x_i; \mu, \sigma, \lambda_s, \lambda_b) = \prod_{i=1}^{N} \left\{ \frac{1}{\lambda_s + \lambda_b} \left[\lambda_s \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} + \lambda_b \cdot \frac{1}{20} \right] \right\}$$

Joint probability distributions for our reference example

Our model has two random variables, X the energy of each event and N the number of events. The joint PDF will hence be:

$$
\begin{aligned}
f_{N,\vec{X}}(n,\vec{x};\mu,\sigma,\lambda_s,\lambda_b) &= f_N(n;\lambda_s+\lambda_b)\cdot f_{\vec{X}}(\vec{x};\mu,\sigma,\lambda_s,\lambda_b) \\
&= \frac{e^{-(\lambda_s+\lambda_b)}(\lambda_s+\lambda_b)^n}{n!}\cdot\prod_{i=1}^{N}\left\{\frac{1}{\lambda_s+\lambda_b}\left[\lambda_s\cdot\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}+\lambda_b\cdot\frac{1}{20}\right]\right\}
\end{aligned}
$$

The likelihood function

Let $f_{\vec{X}}\left(\vec{x};\vec{\theta}\right)$ denote the joint probability distribution of the random variable $\vec{X} = \{X_1, X_2, \ldots, X_n\}$ for a given set of parameters $\vec{\theta} = \{\theta_1, \theta_2, \ldots, \theta_m\}$. Given an observed value of $\vec{X}$ denoted with $\vec{x}$, the function of $\vec{\theta}$ defined by:

$$\mathcal{L}(\vec{\theta};\vec{x}) = f_{\vec{X}}(\vec{x},\vec{\theta})$$

is called the **likelihood function**.

The likelihood function

Let $f_{\vec{X}}\left(\vec{x};\vec{\theta}\right)$ denote the joint probability distribution of the random variable $\vec{X} = \{X_1, X_2, \ldots, X_n\}$ for a given set of parameters $\vec{\theta} = \{\theta_1, \theta_2, \ldots, \theta_m\}$. Given an observed value of $\vec{X}$ denoted with $\vec{x}$, the function of $\vec{\theta}$ defined by:

$$\mathcal{L}(\vec{\theta};\vec{x}) = f_{\vec{X}}(\vec{x},\vec{\theta})$$

is called the **likelihood function**.

The likelihood function is possibly the most important tool for statistical inference:

- think about it as the joint probability of a model, but rather than focusing on the outcome as a variable, consider the outcome as fixed by the experiment and the model parameters as variables
- the likelihood should however not be considered as a probability distribution as the model parameters are not random variables
- if we compare the likelihood function at two points and find that

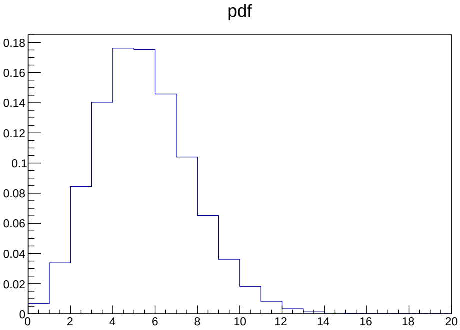
$$P_{\vec{\theta}_1}\left(\vec{X} = \vec{x}\right) = L\left(\vec{\theta}_1;\vec{x}\right) > L\left(\vec{\theta}_2;\vec{x}\right) = P_{\vec{\theta}_2}\left(\vec{X} = \vec{x}\right)$$

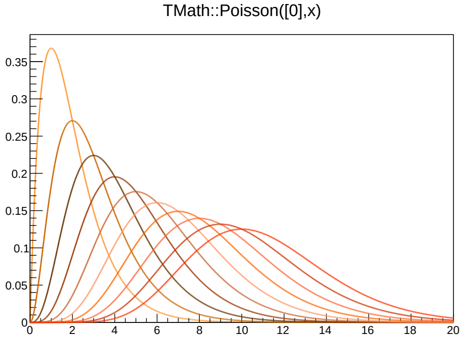
then the observed data are more likely to have occurred if $\vec{\theta} = \vec{\theta}_1$ than if $\vec{\theta} = \vec{\theta}_1$ than if $\vec{\theta}_1$ is a more plausible value for the true value of $\vec{\theta}$ than $\vec{\theta}_2$.

Likelihood function for a Poisson process

Consider a Poisson process and a counting experiment (random variable is N). The expected number of counts in the data set is $\lambda = 5$. The PDF and likelihood functions are

$$f_N(n;\lambda) = \mathcal{L}(\lambda;n) = \frac{e^{-\lambda}\lambda^n}{n!}$$

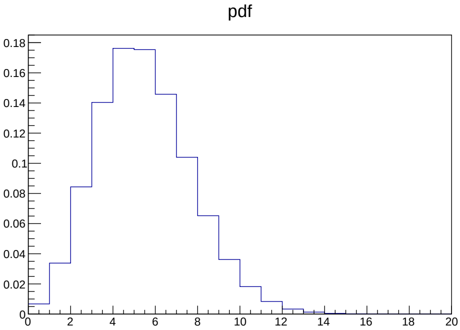


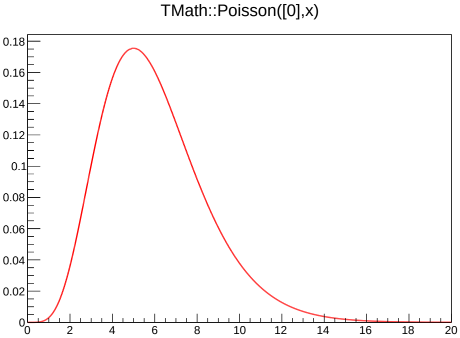


Likelihood function for a Poisson process

Consider a Poisson process and a counting experiment (random variable is N). The expected number of counts in the data set is $\lambda = 5$. The PDF and likelihood functions are

$$f_N(n;\lambda) = \mathcal{L}(\lambda;n) = \frac{e^{-\lambda}\lambda^n}{n!}$$





d

The likelihood principle

The likelihood principle states that all relevant information for inference about $\vec{\theta}$ is contained in the likelihood function for the observed data given the assumed statistical model.

A discussion about the likelihood principle and its statistical/philosophical implications is beyond the scope of this course, but it should not be hard to believe that something connected to the joint probability is all we need for statistical inference

Likelihood function for our reference example

For a given data set in which $N = n$ and $\vec{X} = \vec{x}$:

$$\begin{aligned}
\mathcal{L}(\lambda_s, \lambda_b; n, \vec{x}) &= f_{N,\vec{X}}(n, \vec{x}; \mu, \sigma, \lambda_s, \lambda_b) \\
&= \frac{e^{-(\lambda_s+\lambda_b)}(\lambda_s + \lambda_b)^n}{n!} \cdot \prod_{i=1}^{N} \left\{ \frac{1}{\lambda_s + \lambda_b} \left[\lambda_s \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} + \lambda_b \cdot \frac{1}{20} \right] \right\}
\end{aligned}$$

this is called an "extended unbinned" likelihood function

- it is unbinned in the sense that the energy of each events is used
- it is extended in the sense that the total number of events observed is not fixed and it is described by the Poisson term

Questions?

Point Estimators

- An estimator is a rule for calculating an estimate of a given quantity based on observed data:
- Point Estimators are rules for calculating a single value which is to serve as a "best guess" or "best estimate" of an unknown parameter

Point Estimators

- An estimator is a rule for calculating an estimate of a given quantity based on observed data:
- Point Estimators are rules for calculating a single value which is to serve as a "best guess" or "best estimate" of an unknown parameter
- Examples:
  - Let X be a random variable with a Gaussian PDF with median μ and variance σ^2. Let $x_1, x_2, \ldots, x_n$ be a set of outcomes from independent measurements of X. Estimators for the median and variance of the Gaussian are:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\hat{\sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

Point Estimators

- An estimator is a rule for calculating an estimate of a given quantity based on observed data:
- Point Estimators are rules for calculating a single value which is to serve as a "best guess" or "best estimate" of an unknown parameter
- Examples:
  - Let X be a random variable with a Gaussian PDF with median μ and variance σ^2. Let $x_1, x_2, \ldots, x_n$ be a set of outcomes from independent measurements of X. Estimators for the median and variance of the Gaussian are:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$\hat{\sigma} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{\mu})^2$$

- Some of the properties of an estimator:
  - mean squared error: the average squared difference between the estimator and the parameter it estimates (precision)
  - bias: the difference between the expectation value of the estimator and the true value of the parameter (accuracy)

Maximum likelihood estimators

For a given outcome $\vec{x}$, the maximum likelihood estimator (MLE) of a parameter θ is the parameter value that maximizes $\mathcal{L}(\theta; \vec{x})$ considered as a function of θ with $\vec{x}$ held fixed. We will denote the MLE with $\hat{\theta}(\vec{x})$

- intuitively, the MLE is a reasonable (and the most popular) choice for an estimator.
- The MLE is the parameter point for which the observed sample is more frequent.
- if the likelihood function is differentiable (in $\vec{\theta}$), possible candidates for the MLE are the values of $(\theta_1, \dots, \theta_k)$ that solve
$$\frac{\partial}{\partial \theta_i} \mathcal{L}(\vec{\theta}; \vec{x}) = 0, \qquad i = 1, \dots, k.$$
- if the likelihood is hard to differentiate, the maximum can be found with computational methods. To this purpose, the problem of finding the maximum of the likelihood is converted into finding the minimum of the negative log-likelihood:

$$\hat{\theta} = \left\{ \max_{\theta \in \Theta} \mathcal{L}(\theta; x) \right\} = \left\{ \min_{\theta \in \Theta} -2 \log \mathcal{L}(\theta; x) \right\}$$

MLE for the mean of a Gaussian distribution

$$\mathcal{L}(\mu, \sigma; \vec{x}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$\log\left(\mathcal{L}(\mu, \sigma; \vec{x})\right) = \sum_{i=1}^{N} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right) = -\frac{N}{2}\log\left(2\pi\sigma^2\right) - \sum_{i=1}^{N}\frac{(x_i-\mu)^2}{2\sigma^2}$$

$$\frac{\partial}{\partial\mu}\log\mathcal{L}(\mu, \sigma; \vec{x}) = 0 \longrightarrow \frac{2}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu) = 0 \longrightarrow \sum_{i=1}^{N} x_i - N\mu = 0 \longrightarrow \mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\boxed{\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i}$$

SciPy.org Sponsored By ENTHOUGHT

Scipy.org | Docs | SciPy v1.2.1 Reference Guide | Optimization and Root Finding (`scipy.optimize`)

scipy.optimize.minimize

scipy.optimize.minimize(*fun, x0, args=(), method=None, jac=None, hess=None, hessp=None, bounds=None, constraints=(), tol=None, callback=None, options=None*) [source]

Minimization of scalar function of one or more variables.

Parameters:

fun : *callable*

The objective function to be minimized.

```
fun(x, *args) -> float
```

where x is an 1-D array with shape (n,) and *args* is a tuple of the fixed parameters needed to completely specify the function.

x0 : *ndarray, shape (n,)*

Initial guess. Array of real elements of size (n,), where 'n' is the number of independent variables.

args : *tuple, optional*

Extra arguments passed to the objective function and its derivatives (*fun*, *jac* and *hess* functions).

method : *str or callable, optional*

Type of solver. Should be one of

- 'Nelder-Mead' (see here)
- 'Powell' (see here)
- 'CG' (see here)
- 'BFGS' (see here)
- 'Newton-CG' (see here)

```python
from scipy import optimize

def f(var_par_vector, fix_par1, fix_par2):
    # extract variable parameters from vector
    x = var_par_vector[0]; y = var_par_vector[1]

    return (x - fix_par1)**2 + (y - fix_par2)**2

# starting value of the parameters
par_starting_values = (0,0)
# parameter range (only positive values)
bounds=((0, None), (0, None))

fix_par1 = -2
fix_par2 = 1
results = optimize.minimize(f, par_starting_values, args=(fix_par1, fix_par2), bounds=bounds)

print(results)

print(results.x[0], results.x[1])
```

Many minimizer algorithms available in scipy.
If the function is differentiable, methods based on its gradient are more performing

Exercise 2

1) implement reference model assuming $\lambda_s = 100$ counts and $\lambda_b = 100$ counts (ergo $\lambda_{s+b} = 200$ counts)
2) implement method to generate pseudo-data
3) implement likelihood function (try to share the definition of the PDF's between this and the previous method)
4) analyze single data set:
   - generate pseudo dataset
   - compute MLE for λ_s and λ_b (i.e. minimize likelihood function against λ_s and λ_b)
   - plot likelihood function against λ_s for a fixed value of λ_b
5) analyze 100 data sets:
   - generate data sets and compute the MLE of λ_s and λ_b for each of them
   - store MLE values, plot them in a histogram, and verify that the MLE is unbiased (i.e. the median value converges to the injected value)
   - plot likelihood function against λ_s for a fixed value

macro with solutions: `github.com/mmatteo/TUM-lectures-PH2282`

Binned vs Unbinned likelihoods

For a given data set in which the number of events is $N = n$ and the energy values $\vec{X} = \vec{x} = \{x_1, ..x_N\}$, the likelihoods is:

$$\mathcal{L}(\lambda_s, \lambda_b; n, \vec{x}) = \frac{e^{-\lambda}\lambda^N}{N!} \cdot \prod_{i=1}^{n} f_X(x_i; \lambda_s, \lambda_b)$$

where $\lambda = \lambda_s + \lambda_b$. If we introduce k energy bins, all the events within a bin will give the same contribution proportional to the expectation in each bin:

$$\nu_j = \int_{x \in \text{ j-th bin}} \lambda \cdot f_X(x; \lambda_s, \lambda_b) dx \qquad \text{where } \lambda = \sum_{j=1}^{k} \nu_j$$

Binned vs Unbinned likelihoods

For a given data set in which the number of events is $N = n$ and the energy values $\vec{X} = \vec{x} = \{x_1, ..x_N\}$, the likelihoods is:

$$\mathcal{L}(\lambda_s, \lambda_b; n, \vec{x}) = \frac{e^{-\lambda}\lambda^N}{N!} \cdot \prod_{i=1}^{n} f_X(x_i; \lambda_s, \lambda_b)$$

where $\lambda = \lambda_s + \lambda_b$. If we introduce k energy bins, all the events within a bin will give the same contribution proportional to the expectation in each bin:

$$\nu_j = \int_{x \in \text{ j-th bin}} \lambda \cdot f_X(x; \lambda_s, \lambda_b) dx \qquad \text{where } \lambda = \sum_{j=1}^{k} \nu_j$$

Indicating with $\vec{m} = \{m_1, \ldots, m_k\}$ the number of events per bin with $\sum_{j=1}^{k} = N$, the likelihood becomes:

$$\mathcal{L}(\lambda_s, \lambda_b; \vec{m}) = \frac{e^{-\lambda}}{N!} \cdot \prod_{j=1}^{k} \nu_j^{m_j} = \frac{m_1! m_2! \ldots m_k!}{N!} \cdot \prod_{j=1}^{k} \frac{e^{-\nu_j}\nu_j^{m_j}}{m_j} = \text{const}(\vec{m}) \cdot \prod_{j=1}^{k} \text{Poisson}(m_j; \nu_j)$$

A part from a constant that is fixed given the data set, the binned likelihood is the product of the Poisson probability of observing m_j events in the bin j given and expectation of ν_j

Binned vs Unbinned likelihoods

Extended Unbinned: number of events $N = n$ and energies $\vec{X} = \vec{x} = \{x_1, ..x_N\}$:

$$\mathcal{L}(\lambda_s, \lambda_b; n, \vec{x}) = \text{Poisson}(n; \lambda_s + \lambda_b) \cdot \prod_{i=1}^{n} f_X(x_i; \lambda_s, \lambda_b)$$

Binned: number of events in each bin $\vec{m} = \{m_1, \ldots, m_k\}$

$$\mathcal{L}(\lambda_s, \lambda_b; n, \vec{m}) = \prod_{j=1}^{k} \text{Poisson}(m_j; \nu_j(\lambda_s, \lambda_b))$$

Notes:

- the time to compute (and ergo maximize) the likelihood is proportional to the number of events if the likelihood is unbinned, or to the number of bins if it is binned
- no relevant information is loss if the binning is much smaller than the experimental resolution on X
- MLE can be constructed using different definitions of the likelihood

Questions?