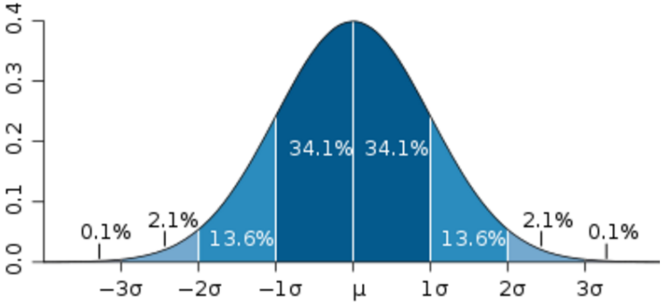
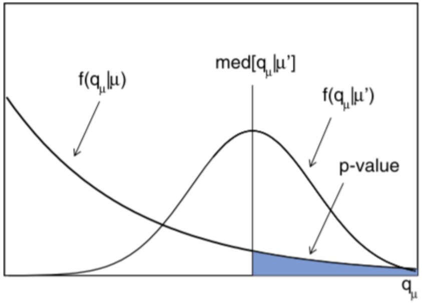




PH2282 part 2: Statistical Methods and Applications to IceCube

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

Hans Niederhausen and Matteo Agostini
TUM - summer term 2019

General Info

Practical information:

- time schedule: every Friday from 10:00 to 13:45
- location: PH 1161
- responsible: Prof. Dr. Resconi (elisa.resconi@tum.de)
- questions: elisa.resconi@tum.de, martin.wolf@tum.de

Structure of the course:

- Dr. Matteo Agostini: introduction to applied statistical methods
- **Dr. Hans Niederhausen: more on statistical methods and applications to IceCube**
- Dr. J. Michael Burgess: fitting a line, applications to astronomy and cosmology
- Dr. Patrick Vaudrevange: cluster analysis and neural networks

4 lecturers discussing statistics from 4 different points of view!
Peek into the analysis methods used in different fields

Introduction to the second block of lectures

About me:

- Postdoc researcher at TUM (E15)
- Member of IceCube Collaboration since 2012
- Office: 2132
- Research: Experimental Astroparticle Physics with Neutrinos (diffuse flux of high energy astrophysical neutrinos, searches of neutrino sources, statistical methods ...)
- Office Hours: flexible, organized through email
- Email: hans.niederhausen@tum.de

About my lectures (upcoming three Fridays):

1) Summary of maximum likelihood methods (point estimation and hypothesis testing)
2) Introduction to IceCube (and relevant physics)
3) Statistical models: describing the detection process
4) Monte Carlo Generation: understanding importance weights
5) Example application: discovering diffuse astrophysical neutrinos
6) Interval estimation and confidence regions
7) Asymptotic properties of maximum likelihood methods
8) Example application: Searching for a point source of neutrinos in the sky (bonus topic)

Lecture organization

IT survey :

- computers?
- linux environment? shell?
- programming experience?
- python? numpy? pylab? scipy?
- C++/ROOT?

Survey on statistic background:

- courses about statistics?
- Gaussian distribution, Poisson distribution?
- likelihoods?
- fitting?
- hypothesis testing?
- confidence intervals?

Motivations and Goals of today's Lecture

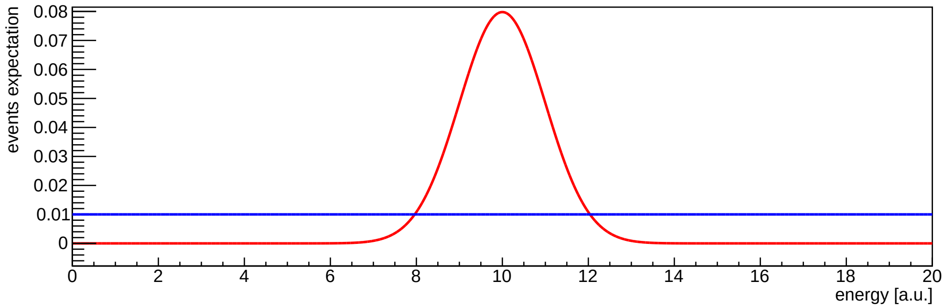
- reinforce and summarize what has been learned so far (statistical models, point estimation and hypothesis testing through maximum likelihood methods)
- learn about Monte Carlo event generation and importance weights
- apply what has been learned to a realistic, recent research problem from IceCube
  "The discovery of a diffuse flux of high energy, astrophysical neutrinos"
- ... and learn about IceCube while doing so.

Many thanks to Dr. Matteo Agostini for introducing the relevant statistics concepts!
(and creating these slides)

Summary of Previous Lectures

- a *statistical model* relates *model parameters* to *random variables*
- a *statistical model* describes the *data generating process*
- the behavior of *random variables* is described by *probability theory*

Summary of Previous Lectures: The Toy Example



The PDF for a generic model in which both $\lambda_s > 0$ and $\lambda_b > 0$ can be written as:

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

where the coefficient $1/(\lambda_s+\lambda_b)$ serves to normalize the PDF.

Summary of Previous Lectures: The toy Example

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:

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

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

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

Summary of Previous Lectures: The Likelihood Function

The data generating process (statistical model) dictates 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**.

Summary of Previous Lectures: The Likelihood Function

The data generating process (statistical model) dictates 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:

- the likelihood function is given by the joint pdf of the observables (considered as constants)
- the likelihood function is a function of the model parameters!
- the likelihood function is NOT a pdf (parameters are not random variables, remember?)
- 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 $\vec{\theta}_2$.

Summary of Previous Lectures: 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

Summary of Previous Lectures: LRT formal definition

In general the two hypotheses tested are expressed as $\boxed{H_0 : \vec{\theta} \in \Theta_0}$ and $\boxed{H_1 : \vec{\theta} \in \Theta_0^c}$ where $\vec{\theta} = \{\theta_1, \ldots, \theta_k\}$ is a vector of model parameters, Θ_0 is some subset of the parameter space allowed for $\vec{\theta}$ and Θ_0^c is its complement. In this case the test statistic can be expressed as

$$t(x) = -2 \log \frac{\mathcal{L}(\hat{\theta}_0; x)}{\mathcal{L}(\hat{\theta}; x)}$$

where $\hat{\theta}$ is the MLE for $\vec{\theta} \in \Theta$ and $\hat{\theta}_0$ is the MLE for $\vec{\theta} \in \Theta_0$.

Considering that the MLE is the value of the parameter that maximized the likelihood in the allowed parameter space, the LRT can be in general expressed as

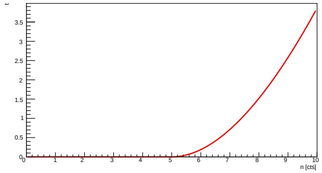
$$t(x) = -2 \log \frac{\sup_{\vec{\theta} \in \Theta_0} \mathcal{L}(\vec{\theta}; x)}{\sup_{\vec{\theta} \in \Theta} \mathcal{L}(\vec{\theta}; x)}$$

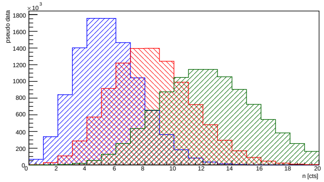
Summary of Previous Lectures: counting with known background

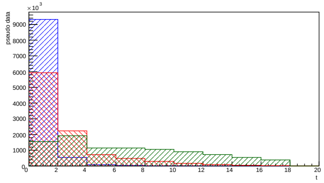
- The experiment still measures a number of counts $N \sim \text{Poisson}(n; \lambda_s + \lambda_b)$ where λ_s is the signal expectation and λ_b is the background expectation
- $\mathcal{L}(\lambda_s, \lambda_b; n) = e^{-(\lambda_s+\lambda_b)}(\lambda_s + \lambda_b)^n / n!$
- hypotheses (no signal vs some signal):
  - $H_0 : \lambda_s = 0, \lambda_b = 5$
  - $H_1 : \lambda_s \geq 0, \lambda_b = 5$
- $t(n) = -2 \log \mathcal{L}(\lambda_s = 0, \lambda_b = 5; n) - (-2 \log \mathcal{L}(\lambda_s = \hat{\lambda}_s, \lambda_b = 5; n))$

$$\hat{\lambda}_s = \begin{cases} 0 & \text{if } n \leq \lambda_b \\ n - \lambda_b & \text{if } n > \lambda_b \end{cases}$$

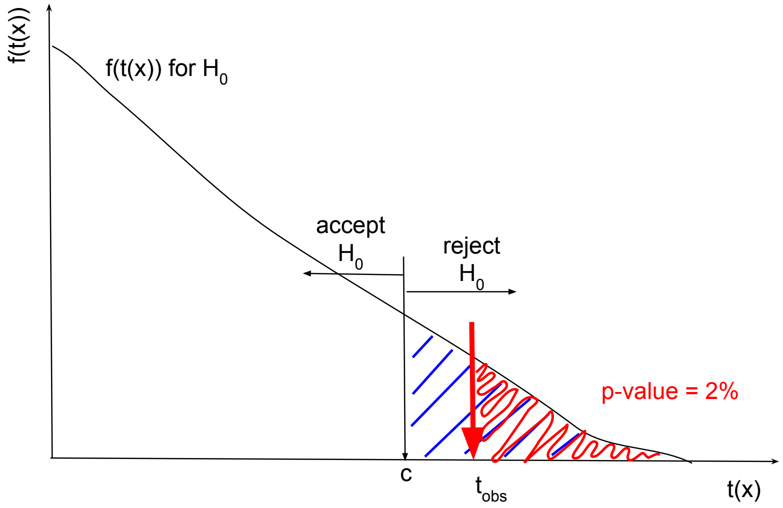
$$t(n) = \begin{cases} 0 & \text{if } n \leq \lambda_b \\ 2\lambda_b - 2n(\log \lambda_b - 1 + \log n) & \text{if } n > \lambda_b \end{cases}$$







Summary of Previous Lectures: p-values



Questions?

Backup

Basic Importance Weights

We often face the problem of evaluating integrals through Monte-Carlo sampling. In some problems naive implementations can turn out to be rather inefficient, i.e. have high variance. Importance sampling is a method to reduce variance at minimial extra computational cost.
Consider the example of calculating the average value $\mu = E_f\left(g(x)\right)$ of function $g(x)$ w.r.t the pdf $f(x)$. A suitable MC estimate can be obtained by sampling f(x) directly $(X_i \sim f(x))$

$$\mu = \int g(x) f(x) dx \tag{1}$$

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} g(x_i) \tag{2}$$

However if g(x) is large in a region where density f(x) is small, we will rarely obtain samples in the region of interest and thus the estimate will be noisy.

Basic Importance Weights

In order to reduce noise, one can instead choose to base the estimate on samples generated from a pdf $h(x)$ with higher density in the region of interest ($x_i \sim h(x)$).

$$\mu = \int g(x)f(x)dx = \int \frac{g(x)f(x)}{h(x)} h(x)dx = E_h(\frac{g(x)f(x)}{h(x)}) \tag{3}$$

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} \frac{g(x_i)f(x_i)}{h(x_i)} \tag{4}$$

The fact that we used $h(x)$ instead of $f(x)$ to estimate μ is corrected by using the importance weights $w_i = \frac{f(x_i)}{h(x_i)}$ in what has now become a weighted average.
A demonstration of this technique can be found in two ipython notebooks (check course repo):

- example 1: weighted_average_ex1.ipynb
- example 2: weighted_average_ex2.ipynb

This technique is used extensively in IceCube (and other HEP experiments) to estimate quantities of interest from Monte Carlo datasets.