KIT-Fakultät für Physik
Institut für Experimentelle Teilchenphysik
Dr. Matthias Schröder

Karlsruher Institut für Technologie

Teilchenphysik 2 — W/Z/Higgs an Collidern

Sommersemester 2019

Specialisations No. 2

Discussion on June 18, 2019

In this exercise, we will study some of the concepts of statistical analysis relevant in particle physics.

The exercise will be a computer exercise, and it will be done during class time ("Präsenzübung"). The exercise runs standalone on a ROOT input file. *Please bring a laptop and make sure **beforehand** that there is a working installation of a recent ROOT6 version* (the exercise has been tested with ROOT version 6.06/06). Also, please set up **beforehand** the environment as described below so that we can start with the exercise right away during class time.

It is encouraged that you work in small groups of up to three persons, and it is sufficient to have one laptop per group.

Exercise 1: Setting up the environment

You need a working installation of a recent ROOT6 version (the exercise has been tested with ROOT version 6.06/06 and Python version 2.7.6). The exercise is performed with the ROOT script `computeLimits.C`, which takes the ROOT file `InputForLimits.root` as input. Download both files from ILIAS into a local working directory.

That's it, you are all set up! All further instructions will be provided during the class.

Exercise 2: Computing exclusion limits

In this exercise, we want to determine exclusion limits on a signal-model parameter. The script `computeLimits.C` contains already all of the tools necessary, but you need to adjust a few things and use the tools correctly.

The input ROOT file contains a histogram `hData`, which represents the observed number of data events in a measurement, sorted in bins of some invariant mass. In addition, there are two histograms `hBkg` and `hSig`, which represent model predictions for a background process and a signal process, respectively. These histograms can be conveniently plotted by executing `computeLimits.C` with

```
root -l computeLimits.C+
```

Plotting of the histograms happens at the beginning of the function `computeLimits()`. `hBkg` is the blue and `hSig` the red histogram.

In the following, we will consider two hypotheses:

- H_0: the background-only hypothesis. The prediction $b_{\text{pred},i}$ for the mean number of events per bin i is given by `hBkg`. There is no free parameter in this model, the prediction is fixed.
- H_1: the signal+background hypothesis. The prediction $b_{\text{pred},i} + \mu \cdot s_{\text{pred},i}$ for the mean number of events per bin i is given by `hBkg` and `hSig`. In addition, H_1 depends on the signal-strength modifier μ.

It is important to realise that $H_0 = H_1(\mu = 0)$.

When comparing the data to the background-only model H_0, it is evident that there is some but no strong indication for a signal being present in the data on top of the background. There is just one bin in the signal region with a significant upward fluctuation above the background.

In absence of a clear signal, we want to use the data to constrain the signal+background model H_1 by setting an upper limit on the signal strength modifier μ. We will do so as follows.

We will use the likelihood ratio test statistic

$$q = q(n_{\text{obs}}, \mu) = -2 \ln \frac{\mathcal{L}_{H_1}(n_{\text{obs}}, \mu)}{\mathcal{L}_{H_0}(n_{\text{obs}})} \,.$$

We will assume that the number $n_{\text{obs},i}$ of observed events per bin is Gaussian distributed around the mean.

- Is this a justified assumption? Which standard deviation of the Gaussian would you use?
- Show that q can in this case be rewritten as

$$q = q(n_{\text{obs}}, \mu) = \chi^2_{H_1}(n_{\text{obs}}, \mu) - \chi^2_{H_0}(n_{\text{obs}})\,.$$

Computation of q is already implemented in the function `getTestStatistic`. Inspect the code and make sure you understand what is happening and why. How is the χ^2 computed?

Now use `getTestStatistic` to compute q for some value of μ, e. g. $\mu = 1$, i. e. compute

$$q_{\text{obs}} = q(n_{\text{obs}}, \mu)$$

with n_{obs} from `hData`. This is already implemented in the main function, it just needs to be uncommented.

- Which value do you find for q_{obs}?
- Suppose the data is different and you find a value smaller than your current q_{obs}. What would be the interpretation: does it mean that the data would be more signal like or more background like?
- Where does the dependency of q on the data come from?

We want to compute the upper limit on μ at 95% confidence level using the $\text{CL}_{\text{s+b}}$ method. The limit μ_{95} is given via

$$\text{CL}_{\text{s+b}} = \int_{q_{\text{obs}}}^{\infty} \mathrm{d}q\, \mathcal{P}(q(\mu_{95})|H_1) = 0.05\,,$$

i. e. $\text{CL}_{\text{s+b}}$ is the p value of the data under the $H_1(\mu_{95})$ hypothesis.

- How is this to be interpreted? What does the p value describe?

In order to compute the p value, we need the pdf $\mathcal{P}(q|H_1)$ of q under the H_1 hypothesis. Computation of $\mathcal{P}$ is already implemented in `getTestStatisticPdf(const double mu, const bool useH1, const int nToys)`. Inspect the function and make sure to understand what is happening.

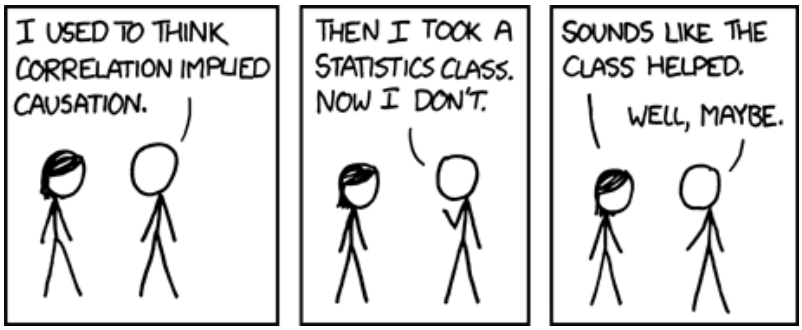
- What are the basic steps to obtain the pdf? What is the toy data needed for? How is it sampled (produced)?
- How does the pdf depend on μ?

Use the already implemented code in the main function to produce and plot the pdf $\mathcal{P}(q|H_1)$ for some value of μ, e. g. for $\mu = 1$. For illustration purposes, overlay also the pdf under the H_0 hypothesis, $\mathcal{P}(q|H_0)$.

- Does $\mathcal{P}(q|H_0)$ depend on μ?
- Why do we use the likelihood ratio test statistic?

Finally, find the upper limit at 95% confidence level on μ using the $\mathrm{CL_{s+b}}$ method.

- Would the 90% confidence-level limit be higher or lower?
- Does the limit depend on the data? Is there a way to get an estimate of what the limit in data would look like if the background-only hypothesis were true?
- How would the procedure be extended to compute the limits with the $\mathrm{CL_s}$ method? What is the motivation?



https://xkcd.com/552/