

## Teilchenphysik 2 — W/Z/Higgs an Collidern

### Sommersemester 2019

#### Exercises No. 3

Discussion on June 4, 2019

---

In this exercise, we will study the topic of triggers, which are a crucial tool for analysis in particle physics. For example, we will learn how trigger efficiencies can be measured.

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 and Python 2 version* (the exercise has been tested with ROOT version 6.06/06 and Python version 2.7.6). It is encouraged that you work in small groups of up to three persons, and it is sufficient to have one laptop per group.

All further instructions will be provided during the class.

#### Exercise 1: Setting up the environment

You need a working installation of a recent ROOT6 and Python 2 version (the exercise has been tested with ROOT version 6.06/06 and Python version 2.7.6). The exercise is performed with the Python program `calculate_efficiency.py`, which takes the ROOT file `histos.root` as input. Download both files from ILIAS into a local working directory. That’s it, you are all set up!

#### Exercise 2: Measuring the trigger efficiency

In this exercise, we want to measure the efficiency of a high-level trigger path that requires the presence of one jet with transverse momentum  $p_T$  above a certain threshold, in our case  $p_T > 500$  GeV. The trigger path is called `HLT_PFJet500`. Since it is the trigger whose efficiency we want to measure, it is generally called *probe trigger*.

We measure the efficiency by using a different trigger as *reference trigger*. In our case, the reference trigger requires the presence of an isolated muon with  $p_T > 20$  GeV and is denoted as `HLT_IsoMu20`. With this, we can define the efficiency  $\epsilon$  as

$$\epsilon = \frac{N(\text{reference} \ \&\& \ \text{probe})}{N(\text{reference})} = \frac{N(\text{HLT\_IsoMu20} \ \&\& \ \text{HLT\_PFJet500})}{N(\text{HLT\_IsoMu20})}, \quad (1)$$

where  $N(\text{reference})$  denotes the number of events in which the reference trigger fired and  $N(\text{reference} \ \&\& \ \text{probe})$  the number of events in which both the reference and the probe trigger fired.

Motivate the definition (1) by discussing the following questions:

- What is the purpose of the reference trigger?
- Why do we require the muon of the reference trigger to be isolated? What could happen if one considers any muon (also non-isolated)?
- Could we use a trigger that fires at random as the reference trigger?
- Could we, instead of using the single-muon trigger, use a trigger that requires the presence of a jet with a lower  $p_T$  threshold than `HLT_PFJet500` as the reference trigger, e. g. a jet with  $p_T > 300$  GeV?

For reasons that will become clear later, we want to measure the efficiency (1) as a function of the jet  $p_T$ :  $\epsilon \rightarrow \epsilon(p_T)$ . Therefore, we first need two histograms: the jet- $p_T$  histograms for all events that fired the reference trigger and for those events that fired both the reference and the probe trigger. These histograms are already implemented in the program `calculate_efficiency.py` (make sure to understand where this happens!). Execute the program with

```
python calculate_efficiency.py
```

This will plot the two histograms. Inspect them and discuss the following questions:

- Why do the entries decrease towards large  $p_T$ ?
- What is the reason for the turn-on at low  $p_T$ ?

Following (1), the efficiency  $\epsilon(p_T)$  is then simply the ratio of the two histograms. This is also already implemented (check where!) but the histogram is not drawn. Adjust the script `calculate_efficiency.py` to also draw the efficiency and execute the script. Inspect the trigger efficiency plot.

- What is the efficiency of the `HLT_PFJet500` trigger path?

- Why is there a smooth *turn-on* region around 500 GeV where the efficiency gradually increases? Why does the trigger not reach its maximum efficiency instantly at  $p_T = 500$  GeV?

In the light of the turn-on feature of a trigger efficiency, consider again the question:

- Could we, instead of using the single-muon trigger, use a trigger that requires the presence of a jet with a lower  $p_T$  threshold than `HLT_PFJet500`, e. g. a jet with  $p_T > 300$  GeV? What condition must be satisfied when a trigger with lower threshold is used as the reference trigger?

### **Exercise 3: Computing the uncertainties of the efficiency**

Have a look at the error bars in the efficiency plot produced in the previous exercise.

- Are they reasonable?
- How are they calculated?

The measured efficiency follows a binomial distribution around the true efficiency  $\epsilon^{\text{true}}$ . You can switch to using binomial uncertainties by adding the `'B'` option to the `TH1::Divide` method.

- How do the error bars change?
- Is this reasonable?

The correct way to quote the uncertainties is to use a confidence interval, derived from the measured  $\epsilon$  (which is an estimator of  $\epsilon^{\text{true}}$ !). There is some discussion in the literature on which confidence intervals to use<sup>1</sup>, which also depends on the use case. The PDG recommends to use the *Clopper-Pearson* interval. We will follow this recommendation. Computation of Clopper-Pearson (and various other) intervals are implemented in the ROOT `TGraphAsymmErrors::Divide` method<sup>2</sup> (Clopper-Pearson is used when specifying the option `'cp'`, which is also the default). Adjust `calculate_efficiency.py` to use a `TGraphAsymmErrors` object for the efficiency with Clopper-Pearson intervals as error bars.

- How do the error bars change? Is this reasonable?

---

<sup>1</sup>See e. g. R.D. Cousins, K.E. Hymes, J. Tucker *Frequentist Evaluation of Intervals Estimated for a Binomial Parameter and for the Ratio of Poisson Means* Nuclear Instruments and Methods in Physics Research A 612 (2010) 388–398, doi:10.1016/j.nima.2009.10.156, arXiv:0905.3831

<sup>2</sup>See <https://root.cern.ch/doc/master/classTGraphAsymmErrors.html>

#### **Exercise 4: Determining when a trigger is “fully efficient”**

Now we will discuss the question of how to define the trigger efficiency in more detail. Of course, we can use the  $p_T$  dependent function and have all information available that is needed in physics analysis. However, often a simplified approach is helpful where we determine the  $p_T$  threshold  $p_T^{\text{thres}}$  above which the trigger has reached its full efficiency. Given the turn-on feature, we can assume the trigger efficiency to remain constant for large  $p_T$  far above the turn-on region. Why? Thus, a typical choice is to define  $p_T^{\text{thres}}$  as that  $p_T$  where the trigger reaches an efficiency of 99% of its maximum efficiency.

Before we determine  $p_T^{\text{thres}}$ , consider the following: We have measured the trigger efficiency in bins of jet  $p_T$ , and thus, there is the danger that statistical fluctuations in one bin could affect our choice of  $p_T^{\text{thres}}$ . In order to correct for binning effects, we first want to fit the efficiency with a continuous function. A suitable function is

$$f(p_T; a_0, a_1, a_2) = \frac{1}{2} \cdot a_2 \cdot \left[ \text{erf}\left(\frac{1}{\sqrt{2}a_0}(p_T - a_1)\right) + 1 \right] \quad (2)$$

with the free fit parameters  $a_i$  and erf denoting the *error function*. Before applying (2), inspect the proposed function:

- How is the error function defined and why is it suitable in this case? (Remember again what the reason for the turn-on feature was!)
- What is the interpretation of the parameters  $a_i$ ?

The fit function (2) is already implemented in `calculate_efficiency.py` as a TF1 object (where?). Use it to fit the efficiency (using the `TGraphAsymmErrors` object with the correct uncertainties, of course!). You can now read off the trigger threshold  $p_T^{\text{thres}}$  from the fitted function. Which threshold do you find? What is the efficiency of the trigger beyond the threshold?

#### **Exercise 5: Measuring the efficiency of a different trigger**

We want to use the developed tools to measure the efficiency of a single-jet trigger with a different threshold. Adjust your `calculate_efficiency.py` to measure the efficiency of the `HLT_PFJet60` trigger (the necessary branch is also in the provided ntuple). What do you observe?