Project 4 : Machine Learning

You may study this assignment using Mathematica or Python. Take your pick, but briefly motivate your choice in the assignment! Hand in the corresponding notebooks (.nb or .ipynb formats) before the deadline: **November 2, 9:00**.

We would also appreciate to get some feedback from you about the assignments, your struggles, in particular how much time you have invested in solving them. You can add comments on that at the end of your notebook.

Particle identification using machine learning techniques!

In this assignment, you will be confronted with Monte Carlo data simulated for the future PANDA experiment [1]. This experiment studies the interaction of antiprotons with protons to study how the fundamental building blocks of nature, quarks and gluons, form building blocks of matter such as the proton or neutron. To prepare such an experiment, a detailed software package has been developed that provides the tools to simulate the response of the foreseen experimental setup to the emission of particles that are expected to be produced in antiproton-proton collisions.

Figure 1 depicts the experimental setup that is foreseen. The setup is composed of several components, whereby each component is capable to measure the properties of part of the radiation, *i.e.* particles, that is emitted after the collisions. In this assignment, you will only focus on pre-processed data that are produced by the so-called electromagnetic calorimeter (indicated by “EM Calorimeter” in Fig. 1), abbreviated with EMC. The EMC is designed to measure the scattering angles, and, the energy of high-energetic (~GeV) photons originating from the antiproton-proton interaction point. Other detector components within the setup are blind to those particles. The EMC is composed of about 15.000 crystals made of lead-tungstate each with a dimension of about 25x25x200 mm^3. When a particle, such as a photon, originating from the antiproton-proton interaction point hits the EMC, it will create an electromagnetic shower composed of a massive number of secondary photons, electrons, and positrons covering several adjacent crystals. The energy deposit in each crystal will generate scintillation light which intensity is registered with photo-diodes converting the information into an electric signal. In short, the energy deposit in each crystal is measured and low-level algorithms are used to find clusters of firing and adjacent crystals and the properties of these clusters are further processed to reconstruct the properties of the incident candidate photons hitting the EMC. Figure 2 shows an example spectrum of the formation of clusters for five photons hitting the forward part of the calorimeter. One can clearly observe several clusters of adjacently firing crystals.

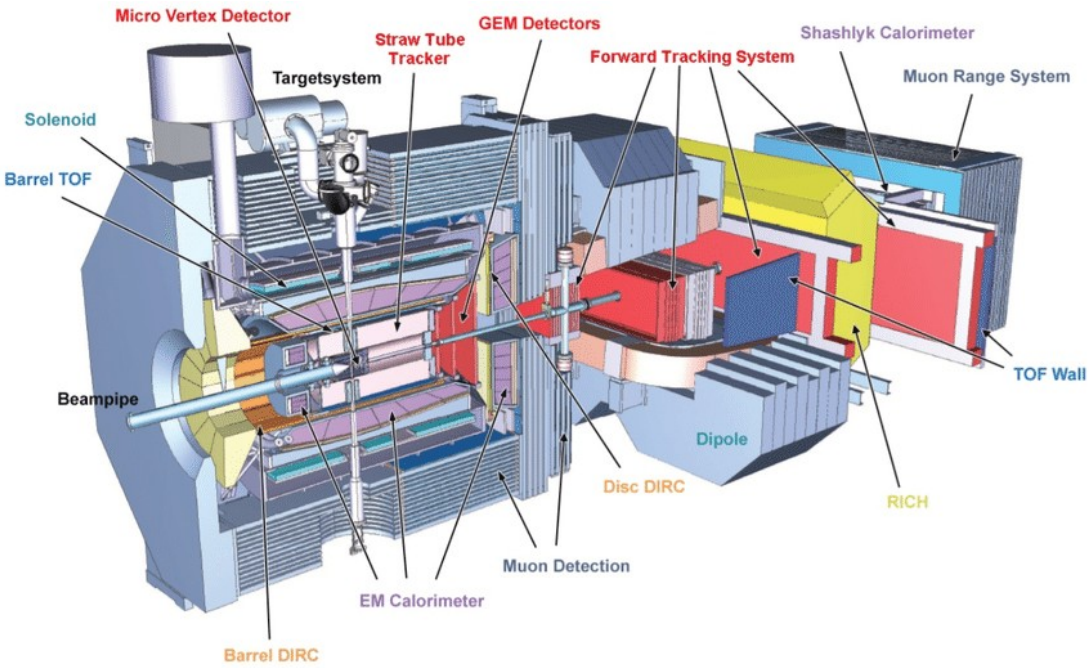

Figure 1: A cross section of the PANDA detector. The antiproton beam (from left) hits a proton target. Scattering angles and momenta of produced particles are measured using various detector components. The electromagnetic (EM) calorimeter is indicated in purple.

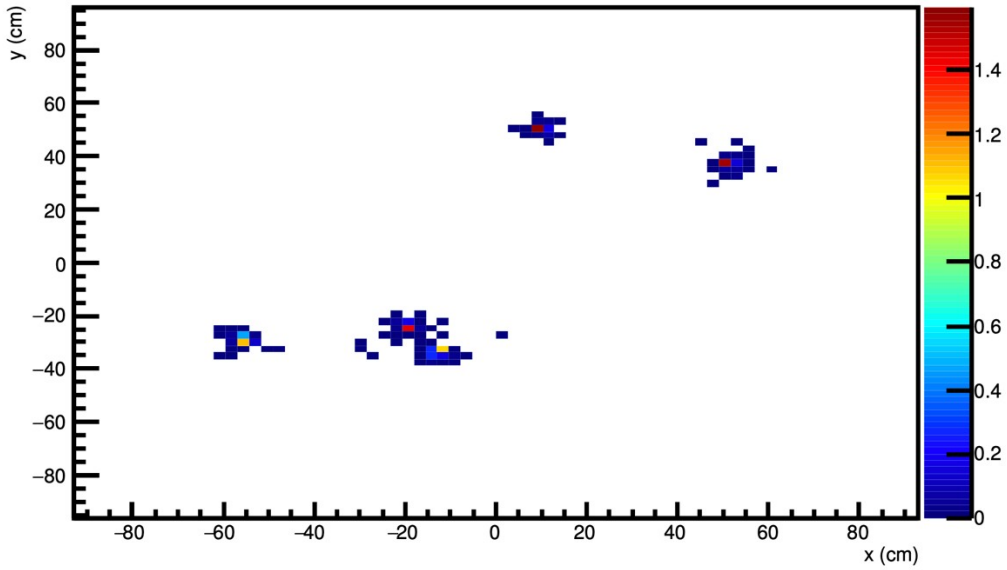

Figure 2: Cluster formation for five high-energetic photons hitting the forward part of the EMC. Each bin represents one crystal whereby the color intensity corresponds to the energy deposit (GeV).

One of the challenges lies in suppressing background events and to ensure that only photons are selected. One of the background sources stems from neutrons. Also these particles are electrically neutral and the EMC has a large chance to produce signals as a response to these type of particles as well, thereby mimicking photons. To suppress this type of background, one can exploit the measured properties of the reconstructed clusters in combination with machine learning techniques. In this assignment, you will apply machine learning techniques to study Monte Carlo simulated data that are summarized in two files. The first file (*emc_gam.txt*) has been generated by simulating the response of the

EMC to ~50.000 photons with momenta ranging from 1-5 GeV/*c*. The second file (*emc_neutron.txt*) corresponds to the response of the EMC to neutrons with momenta in the same momentum range. For both files, you find for each impinging photon/neutron the measured properties of the EMC, separated by commas. The rows, therefore, correspond to events, *e.g*. instances, and the column the various measured properties, *e.g*. features, of the EMC cluster related to the impinging particle, either photon or neutron. A short description of each feature can be found below:

Th (theta): The polar angle of the cluster (in degrees). This parameter is obtained from an energy-weighted average of the positions of the crystals that take part in the cluster.

Ph (phi): The azimuthal angle of the cluster (in degrees). Similar to theta, also this parameter is calculated from an energy-weighted average of the positions of the crystals involved in the cluster.

E (energy): The total deposited energy of the cluster (GeV). It is the sum of deposited energies found in each crystal.

NrHits (number of hits): The number of firing crystals that take part in the cluster.

NrBumps (number of bumps): Number of local maxima in energy within the cluster.

E1 (energy of detector 1): The deposited energy of the central crystal (1) in the cluster (GeV). It corresponds to the energy of the crystal with the highest deposited energy among all firing crystals in the cluster.

E1E9: The ratio between E1 and E9. For E1, see above. The parameter E9 corresponds to the sum of deposited energies of the 9 neighboring crystals surrounding the central detector. Hence, it corresponds to the total energy of the first ring around the center of the cluster. Note, however, that E1E9 is a dimensionless parameter being the ratio of E1 and E9.

E9E25: The ratio between E9 and E25. For E9, see above. The parameter E25 corresponds to the sum of deposited energies of the second ring of crystals (25 in total) with respect to the center (1). Also this feature has no dimension being a ratio of E9 and E25.

Z20: The so-called “Zernike” moment with subscript 2-0. It is one of the many parameters that contain information about the shower shape of the cluster. In general, Zernike polynomials are a set of orthogonal complex polynomials defined on a unit disk named after Fritz Zernike (yes, *our* Zernike) that can be used to describe the shape of an optical lens. Although their form originates from the study of circular wavefronts, they can also be used to calculate moments based on the energy distribution within a cluster. For mathematical details, I refer to [2].

Z53: Another Zernike moment with subscript 5-3. Similar to Z20, it reflects a complementary property of the shape of the energy distribution of the cluster [2].

LatMom (lateral moment): The lateral moment is directly related to the expectation value $\langle r^2 \rangle$ of the cluster, whereby r is the lateral distance of a crystal with respect to the center of the cluster. The average is taken by summing over all participating crystals with a weight factor that is related to the deposited energy.

The end goal of this assignment is to work-out a machine-learning based method, *i.e.* classifier, to suppress the neutron background while keeping as much as possible the signals of interest (photons). More quantitatively, the aim is to optimize the so-called figure-of-merit (FOM) or performance metric defined by

$$FOM = \frac{S}{\sqrt{S+B}},$$

whereby S is the number of selected signal (photon) events and B the number of remaining background (neutron) events after classification. This figure-of-merit corresponds to the statistical significance of your selected data sample. It is the same as stating that the statistical uncertainty of S is estimated as $\sqrt{S+B}$. Hence to minimize the relative statistical uncertainty with respect to S, one maximizes the FOM.

Tasks:

- Read the two data files and visualize the various features comparing the responses for photons with that of neutrons. For those using Python, I strongly advice to make use of the *pandas* library [3]. For those using Mathematica, the easiest is to make use of *Dataset* [4]. Conclude from your observations and discuss and motivate which features you believe are likely to be the most powerful ones (and which not). Also study the correlations among the various features. What can you learn from that?

- The next step is to setup the basics of the machine learning code. Develop a piece of code that returns the output of a supervised classifier, preferably also the FOM, whereby one can "easily" select the type of classifier (f.e. kNN, random forest, MLP, …) and the list of features to be used as input. If you use Python, make use of the *sklearn* library [5]. In Mathematica, make use of the functionalities provided by *Classify[…] [6]*.

- Pick the features that you believe are the most powerful ones (from the visualization study), and optimize the internal parameters of a classifier you believe could be useful for this application (motivate your choice!). For instance, if you selected kNN, find the optimum "k" and the signal probability for the selected features that maximizes the FOM. Divide up the data sample in 50% training and 50% test and also consider to apply cross validation in your analysis. Do you see the effect of over- or underfitting?

- Study the classifier outputs and FOM by varying the list of input features to the classifier. Which of the features does the classifier believe to be the most powerful ones, which are redundant or do not have any effect in the end result? Discuss your observations.

- *Extra*: Compare the outcome of two different classifiers with each other. In the extreme case, you may even challenge yourself by comparing a supervised with an unsupervised method and compare their performances. Do you see differences in classification power, if so, could you explain why?

References

[1] https://panda.gsi.de

[2] https://en.wikipedia.org/wiki/Zernike_polynomials

[3] https://pandas.pydata.org

[4] https://reference.wolfram.com/language/ref/Dataset.html

[5] https://scikit-learn.org/stable/index.html

[6] https://reference.wolfram.com/language/ref/Classify.html