



Project Report

2021 DESY (Online) Summer Student Programme

by

Raïssa Costa Barroso

at the

Deutsches Elektronen-Synchrotron (DESY)
Platanenallee 6
15738 Zeuthen
Germany

under the supervision of

Dr. Jason J. Watson

August 2021

Contents

1	Introduction	2
2	Motivation and Background	4
2.1	The Data	4
2.2	Pipelines	4
3	The DBSCAN Algorithm	6
3.1	Concepts and Description	6
3.2	Other Algorithms	8
4	Method and Results	9
4.1	Parameter Sampling	9
4.1.1	Defining Optimal Parameters	9
4.1.2	Finding Optimal Parameters	10
4.2	Knee Curve	11
4.3	Comparisons	14
4.3.1	Sampling vs. Knee Curve	14
4.3.2	Proton vs. Gamma showers	14

1 Introduction

Cherenkov Telescope Array

The Cherenkov Telescope Array (CTA) is an ongoing project which aims at detecting high energy gamma rays coming from space and reaching the Earth's atmosphere. In fact, high energy gamma rays interact with the Earth's atmosphere, producing particle showers. These particles have a speed greater than the speed of light in the air, leading to the emission of Cherenkov light. CTA's telescopes are specifically designed to detect such events. Indeed, CTA is composed of different sized telescopes covering a wide energy range: large-sized telescopes (LSTs) with a low energy sensitivity between 20 and 150 GeV, medium-sized telescopes (MSTs) covering energies going from about 150 GeV to 5 TeV and small-sized telescopes (SSTs) with a high energy sensitivity between a few TeV and 300 TeV [1]. The telescopes will be located both in the Southern Hemisphere in Chile, and in the Northern Hemisphere in La Palma, totalling 118 telescopes!

It is an impressive enterprise, not only from a technical point of view. CTA will address multiple open questions in astrophysics and broadly speaking in high energy physics. Gamma rays are emitted by a number of different sources such as black holes, pulsars and binary systems, to cite a few [1]. The precise study of the gamma rays emitted by these objects, provides us with crucial information for better understanding the physical processes at play.

Project

Now, it goes without saying that there is a long way between the detection of Cherenkov shower and the extraction of relevant information. Precisely, the process of going from raw data to the reconstruction of important physical quantities of the progenitor astrophysical particle (e.g. energy, direction) is what is called a pipeline. The goal of this project is to explore a new

1. INTRODUCTION

pipeline, using a different representation of the data in terms of photon lists and in particular to investigate the possible application of machine learning algorithms to the analysis of the data.

Organisation

This report is organised in the following way. Firstly, we introduce the general elements necessary to put this project into context. We then move on to general considerations about DBSCAN, the algorithm we used. Finally, we describe the methods we devised to determine the optimal DBSCAN parameters to be used when studying proton and gamma showers.

Acknowledgements

Carrying out this project remotely was not the easiest of tasks. I would like to thank my supervisor for his patience, dedication and availability.

2 Motivation and Background

2.1 The Data

Throughout this project we use simulated data. In order to produce this data, we use the `sstcam-simulation` Python package, which simulates the SST camera definition and allows to assess the performance of the camera. In our case, we extract a photon list from the simulation. It is important to stress here that up to this point, our data consists just of Cherenkov photons. Then by assuming a Poisson distribution with an average which corresponds to the photon arrival rate per pixel, we add night sky background (NSB) photons.

In Fig. 2.1, the data is represented in three-dimensional space. The position of a photon in the x-y plane corresponds to the position at which it reached the camera (which is flat). The position of a photon in the time axis corresponds to the time at which it reaches the camera.

2.2 Pipelines

There is a long way from the detection of a Cherenkov shower to the extraction of interesting quantities for the study of the physical processes at the origin of the emission of high energy gamma rays. In fact, after being detected, a Cherenkov shower is shaped by the electronics of the system, digitised and then stored as waveforms of 128 ns long with 1 ns sampling. In the traditional pipeline, from these raw waveforms one typically generates images of these Cherenkov showers. Then from the analysis of these images one reconstructs the main properties (such as energy, direction, type) of the progenitor astrophysical particle.

In the pipeline we investigate, however, we do not go through the process of generating two-dimensional images of charge and arrival time. Instead, we

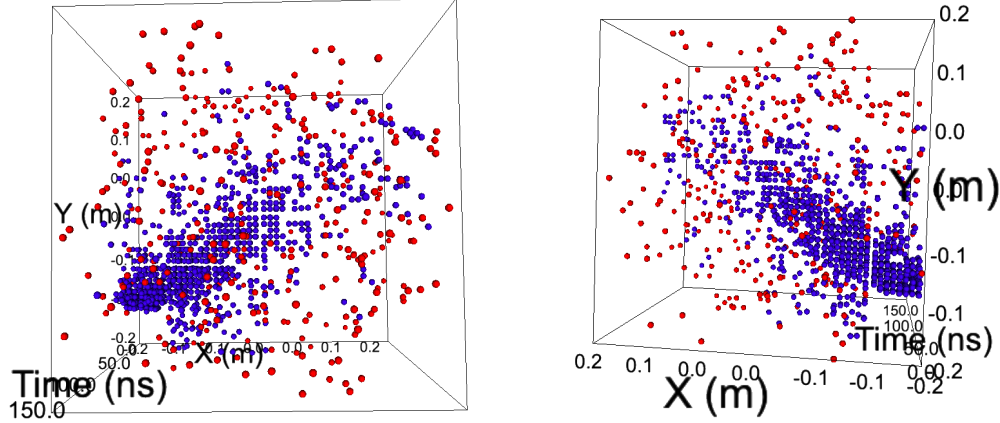


Figure 2.1: Proton cosmic rays Cherenkov shower (blue) and NSB (red) simulation.

use photon lists. Photon lists are a different representation of the showers. In fact, they are generated by the application of algorithms such as the Non-Negative Least Squares (NNLS) algorithm to the traditional waveforms. This means we only keep potentially useful events, so we keep all their information. Since a large amount of useless information is discarded from the start, we can afford with photon lists to keep more detailed information, in particular concerning time. Moreover, this different representation is well suited for the application of machine learning algorithms for discriminating between different showers and noise.

3 The DBSCAN Algorithm

Machine learning algorithms have proved to be efficient solutions when it comes to data analysis. Some examples are applications to classification, regression and clustering problems. In our particular case, we are looking for a clustering algorithm. In fact, we want to discriminate between photons related to the Cherenkov shower, the physical event we are interested in, and photons associated to the background. As showed in Fig.2.1, a Cherenkov shower corresponds to a higher density of photons, or in other words to a cluster of photons. This observation motivates the use of the DBSCAN algorithm [2] which is available in the scikit-learn package [3], a module which provides a number of machine learning tools in Python.

3.1 Concepts and Description

The core idea of the DBSCAN algorithm is to work with a carefully defined density based notion of cluster. It is somewhat intuitive to think of clusters in terms of density: if there are many particles localised in a region of space (and few others spread out) we can easily group them together.

In order to understand the algorithm, we have to formalise this intuition. In particular, we need the notions of:

- *Neighbourhood*: a neighbourhood $N_{eps}(p)$ is the set of points within a radius eps of point p .
- *Directly density-reachable*: a point p is directly reachable from a point q if p is in the neighbourhood of q , that is $p \in N_{eps}(q)$, and the $N_{eps}(q)$ has at least a number of *min_samples* elements.
- *Density-reachable*: two points p, q are density-reachable if one can reach p from q with a chain of directly density-reachable points.

3. THE DBSCAN ALGORITHM 3.1. CONCEPTS AND DESCRIPTION

- *Density-connected*: two points p and q are said to be density-connected if there exists a point o such that o is density-reachable from p and from q with respect to eps and $min_samples$.

The rigorous definitions can be found in [2]. Let us stress here that all these notions depend on one or both of the parameters eps and $min_samples$. Thus these are crucial parameters for DBSCAN. To be precise, when we write (directly) density-reachable or density-connected in what follows, we mean with respect to some eps and $min_samples$.

The above notions lead us to the definitions of cluster and noise used by DBSCAN:

- *Cluster*: If D is a database, then the cluster C with respect to eps and $min_samples$ is a subset of D such that: (i) if a point p in C is density-reachable from another point q , then q also belongs to C ; (ii) all pairs of points in C are density-connected.
- *Noise*: noise is the set of points which do not belong to any cluster in the database.

With these definitions in hands, we can already get an idea of how DBSCAN will tackle the problem of finding clusters. Starting from a point p , DBSCAN will look for density-reachable points from p . This step will lead to the identification of a cluster, provided that p is indeed part of a cluster. If this is the case, then DBSCAN will move on to another point outside of the first cluster. If it is not the case, then DBSCAN will explore any other point of the database.

A final remark on DBSCAN concerns its classification as a machine learning algorithm. In fact, the general representation we have of machine learning is usually restrained to supervised learning. In this case, we think of machine learning as being a black box which after comparing its own results to expected outputs coming from a large database understands how to return desired answers. There is, however, another type of learning, so-called unsupervised learning. DBSCAN falls into this category since it creates its own representation of the data it is provided with. We do not give any information about the input and the algorithm figures out its underlying structure.

3.2 Other Algorithms

It is worth mentioning where DBSCAN stands in the landscape of available algorithms. One of the advantages of DBSCAN is that no prior information on the data is required. In particular, as opposed to another popular clustering algorithm, K-Means, the number of clusters does not have to be known in advance and clusters can have different sizes. Nonetheless, there is no obvious way of choosing the *eps* and *min_samples* parameters which do depend on the input. Finally, in terms of complexity, DBSCAN scales as $\mathcal{O}(n \log n)$ [2].

4 Method and Results

As briefly discussed in the previous section, the most important parameters in DBSCAN are the parameters *eps* and *min_samples* because a cluster is defined with respect to *eps* and *min_samples*. Therefore, if we want the algorithm to properly distinguish between noise and Cherenkov showers we have to correctly tune these parameters. We have tried two approaches for finding optimal parameters, which we describe in this section.

4.1 Parameter Sampling

4.1.1 Defining Optimal Parameters

In order to find the optimal parameters *eps* and *min_samples* which best identify the shower from the noise, we have to define a criterion which quantifies what we mean by a good discrimination between signal and noise. There is no straightforward way to decide which criterion to use and this choice ultimately depends on the information we want to extract from the data.

The very first criteria we tried consisted in comparing the output of the algorithm against the known nature of a given signal. Since the data we use comes from simulations, we know if a photon comes from the simulation of a Cherenkov shower or from the NSB. So we can compute for example the number of true positives, that is the number of photons which were identified as Cherenkov photons by the algorithm and did indeed come from the simulation of a Cherenkov shower. Similarly, we can define the number of false positives, which were deemed positive but were actually NSB photons in the simulation, and so on. We encountered two main problems with this approach:

- We want to be able to choose a set of parameters which does not mistakenly consider some NSB photons to be Cherenkov photons. This information is encapsulated in all four criteria: number of true positives, true negatives, false positives and false negatives. However, it is not clear how to combine the information provided by the different criteria.
- These criteria are not sensitive to the brightness of the shower. For instance, a dimmer shower might get a lower score even if more photons were correctly identified. So we cannot use this scoring system to compare the performance of the algorithm when applied to showers with different brightness.

Hence, a better scoring system would consist in a single number which carries all the information and takes brightness into account. A statistical tool which corresponds to what we are looking for is the root mean squared. So we define:

$$\text{RMSE} = \sqrt{\sum_i \frac{(b_i - \beta_i)^2}{N}}, \quad (4.1.1)$$

where the indices i go through all the photons in the photon array. We define b_i to be equal to 1 if the photon i was considered to be a Cherenkov photon by DBSCAN and 0 otherwise. Similarly β_i is equal to 1 if this photon was originally a Cherenkov photon in the simulation and 0 otherwise. N is the brightness of the shower, that is the total number of incoming photons. With this formula, we would expect the RMSE value to be minimised when there is a good agreement between the true nature of the photon in the simulation and DBSCAN. Whenever a photon is misidentified (as a Cherenkov or NSB photon) a price of $1/N$ has to be paid. Therefore, the problem of finding an optimal parameter now translates into finding the set of parameters which minimise the RMSE value.

4.1.2 Finding Optimal Parameters

Now that we have a scoring system to measure the performance of the algorithm we can try different set of parameters and find the best *eps* and *min_samples* values. A first approach is to brute force the problem and try many sets of parameters. We essentially create a matrix whose entries are the RMSE values for a set (*eps*, *min_samples*) of parameter values. We

represent this matrix in the form of a two-dimensional plot, where each color represents an RMSE value. We generate such plots for one thousand showers with different brightness and then average out the results between showers within some range. We follow this procedure for both gamma-ray and proton cosmic-ray showers (see Fig.4.1 and Fig.4.2). Note that the values of *eps* and *min_samples* samples are somewhat arbitrary¹.

Based on Fig.4.1, we notice that even if the minimum RMSE value does not correspond to exactly the same set of parameters at different energies, optimal *eps* and *min_samples* do have similar values. This means one could choose a global set of parameters and apply it to showers with different brightness. This solution would not be optimal in all cases, but would not be far from optimal. For instance, one could choose the set (*eps* = 0.29, *min_samples* = 10). Similar observations apply to Fig.4.2. In addition, it worth noting that from Fig.4.1 and Fig.4.2 it is not clear if there is a significant difference between optimal parameters for gamma-ray or proton cosmic ray showers.

4.2 Knee Curve

Another approach, somewhat specific to DBCAN, consists in generating a knee curve by computing the 4-distance, the distance of a point to its fourth nearest neighbour, for all points in the database. Then ordering points in decreasing order of 4-distance, one can generate a plot and then identify the knee of the curve (see Fig.4.3). Following this heuristic approach described in [2], the optimal *eps* and *min_samples* values correspond to the y and x-coordinate of the knee point respectively.

The major shortcoming of this method is the fact that the knee point has to be identified by eye, which makes this approach hard to automatise. In fact, the standard way of formalizing the idea of a knee point is to phrase the problem in terms of the curvature. In mathematics, the curvature of some real function f is defined as:

$$\kappa(x) = \frac{f''(x)}{[1 + f'(x)^2]^{3/2}}. \quad (4.2.1)$$

¹In reality, we can get some intuition from the definition of these parameters. In practice, we have started from a larger sampling space and then focused it in the area which carried more interesting information.

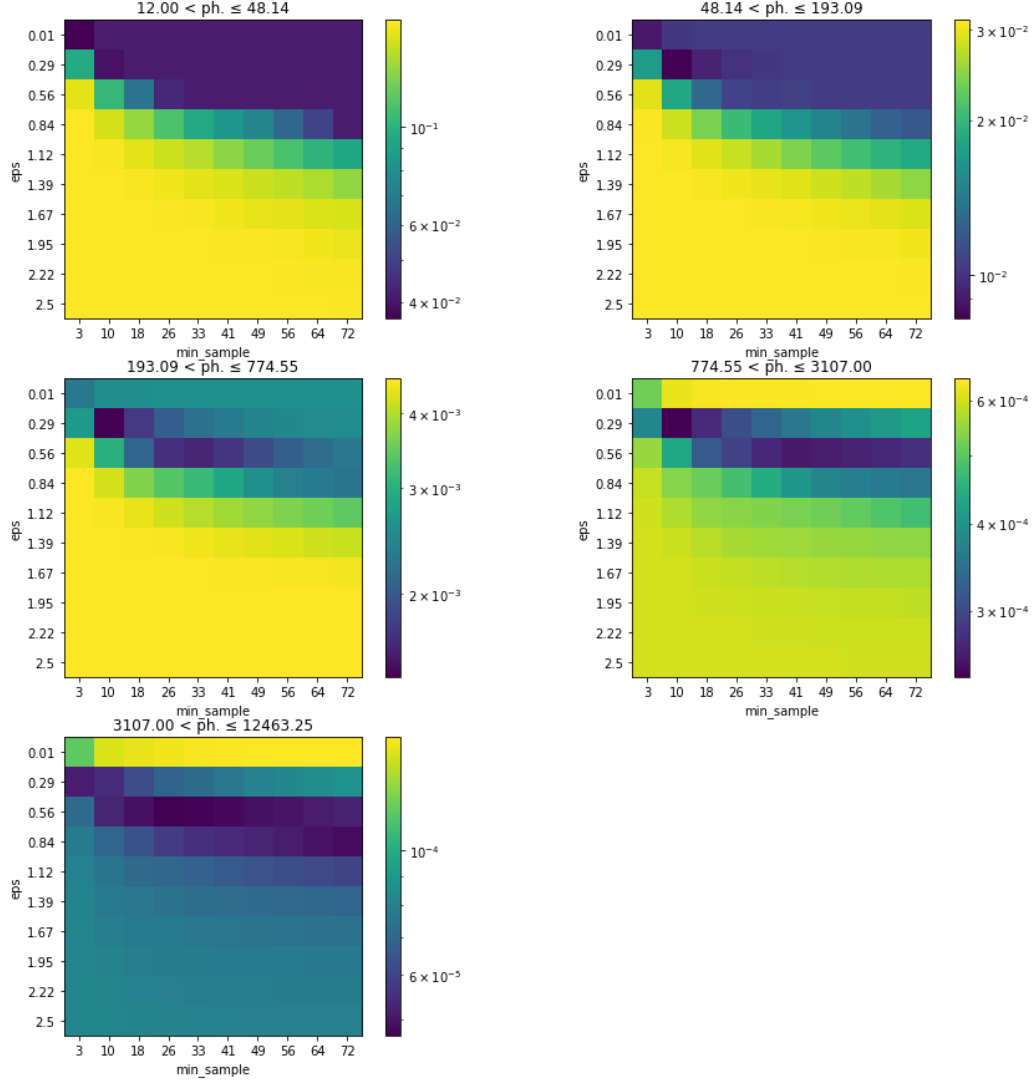


Figure 4.1: Two-dimensional plot of the RMSE values (colormap), as a function of the eps and min_samples DBSCAN parameters for proton cosmic-ray showers. Each plot corresponds to the average of the results obtained for multiple showers within a different range of brightness (in number of photons).

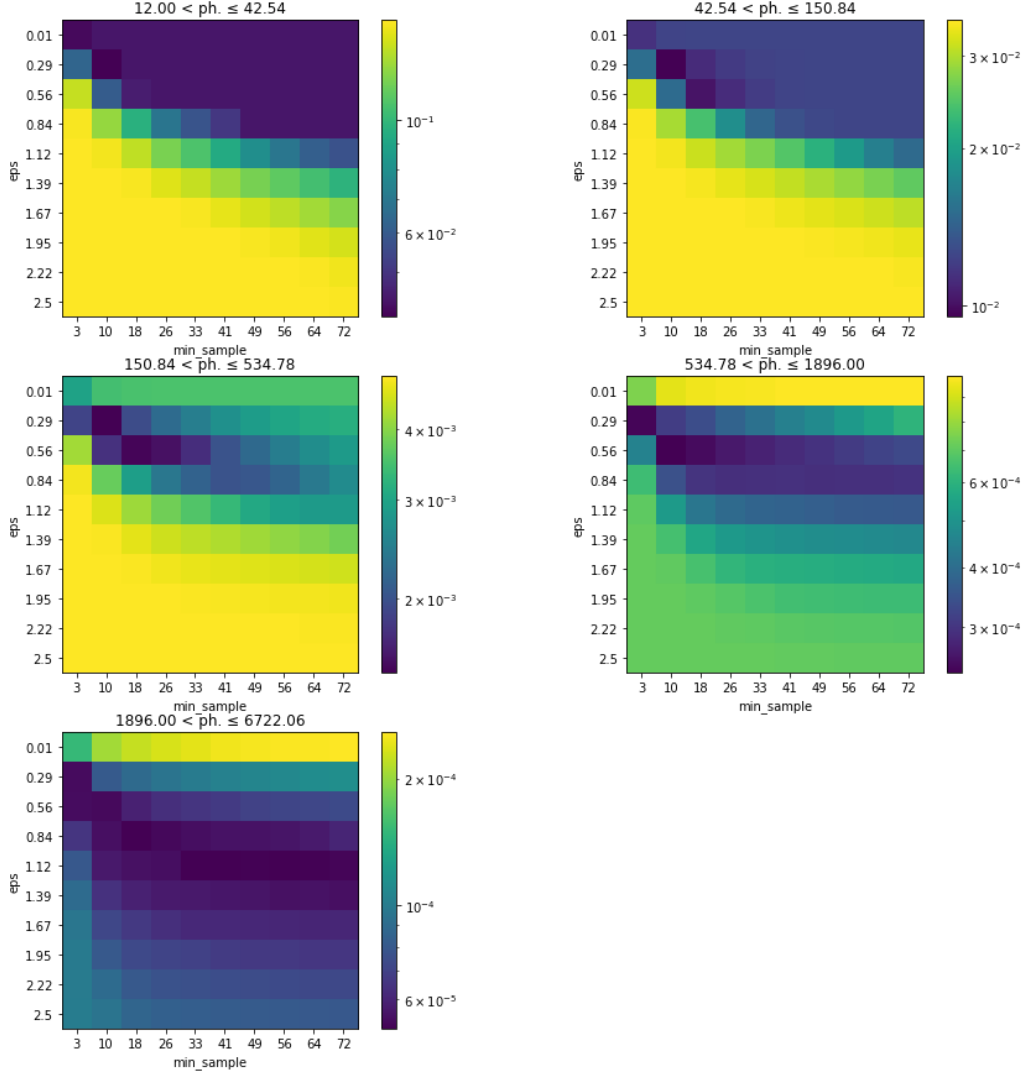


Figure 4.2: Two-dimensional plot of the RMSE values (colormap), as a function of the eps and min_samples DBSCAN parameters for gamma-ray showers. Each plot corresponds to the average of the results obtained for multiple showers within a different range of brightness (in number of photons).

One would then claim that a knee point corresponds to the point of maximum curvature. So a knee point would be where the derivative of the curvature cancels out

$$\kappa'(x) = \frac{f'''(x)[1 + f'(x)^2]^{3/2} - 3f''(x)^2 f'(x)[1 + f'(x)^2]^{1/2}}{[1 + f'(x)^2]^3} = 0, \quad (4.2.2)$$

which amounts to solving the simplified equation:

$$f'''(x)[1 + f'(x)^2] - 3f''(x)^2 f'(x) = 0. \quad (4.2.3)$$

Unfortunately, the curves are well fitted by a rational function,

$$f(x) = \frac{a}{x + b} + c, \quad \text{with } a, b, c \text{ fit paramaters}, \quad (4.2.4)$$

which means this approach fails and we cannot determine the precise location of the knee of the curve by solving Eq.(4.2.3).

4.3 Comparisons

4.3.1 Sampling vs. Knee Curve

The first comparison we want to make is between the two methods described above. In order to check their compatibility, we add the point corresponding to optimal parameters found with the knee curve method to the two-dimensional plot (see Fig.4.4).

If the two sets were to be compatible, the additional point corresponding to the parameters found with the knee curve approach, should be in a region where the RMSE is minimised. This is, however, not the case. In fact, the RMSE formula takes the NSB data into account, providing a more reliable scoring system for our specific database. As opposed to the knee curve method, which relies on the information available to DBSCAN. Therefore, we prefer the more reserved sampling optimum values over the knee curve optimum values. In addition, the knee curve approach has the inconvenient of being hard to automatise.

4.3.2 Proton vs. Gamma showers

Another point we want to further investigate is the difference between proton cosmic ray showers and gamma ray showers' optimal *eps* and *min_samples*

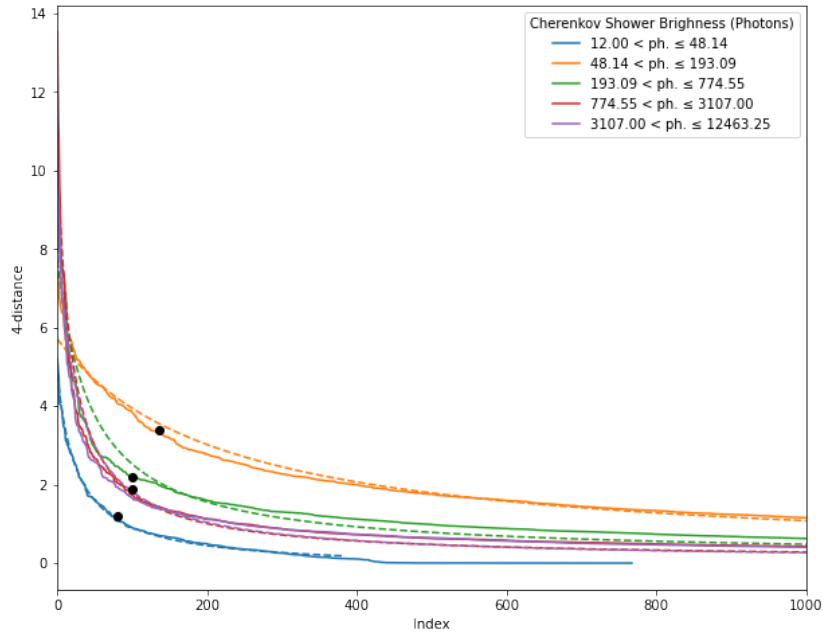


Figure 4.3: Knee plot for proton cosmic ray showers. Each curve corresponds to the accumulated data from many showers within the same brightness range. The dashed lines correspond to a fit by a rational function Eq.(4.2.4). In black are the knee points for proton cosmic ray showers.

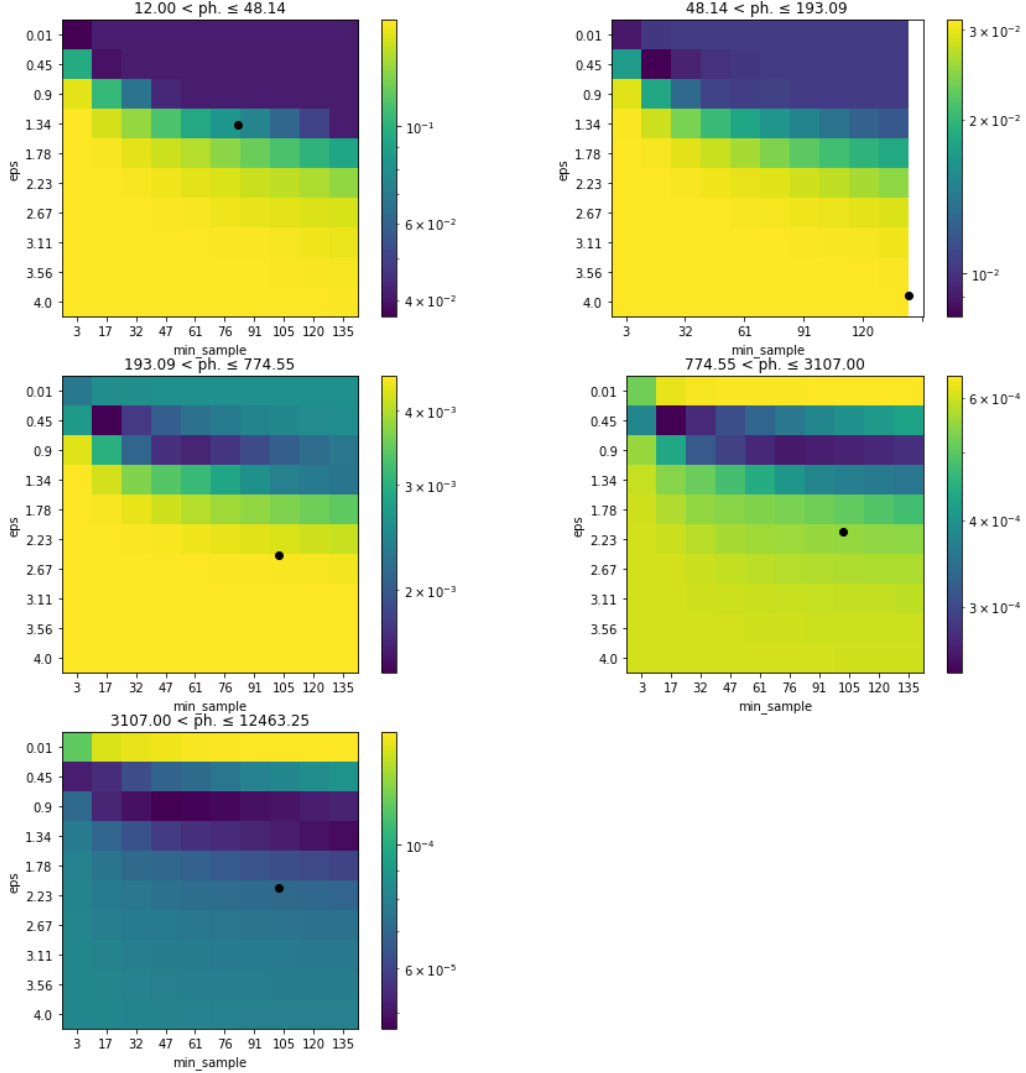


Figure 4.4: Two-dimensional plot of the RMSE values (colormap), as a function of the ϵ and min_samples DBSCAN parameters for proton cosmic-ray showers. Each plot corresponds to the average of the results obtained for multiple showers within a different range of brightness (in number of photons). The black point corresponds to the optimal ϵ and min_samples values found with the knee curve method.

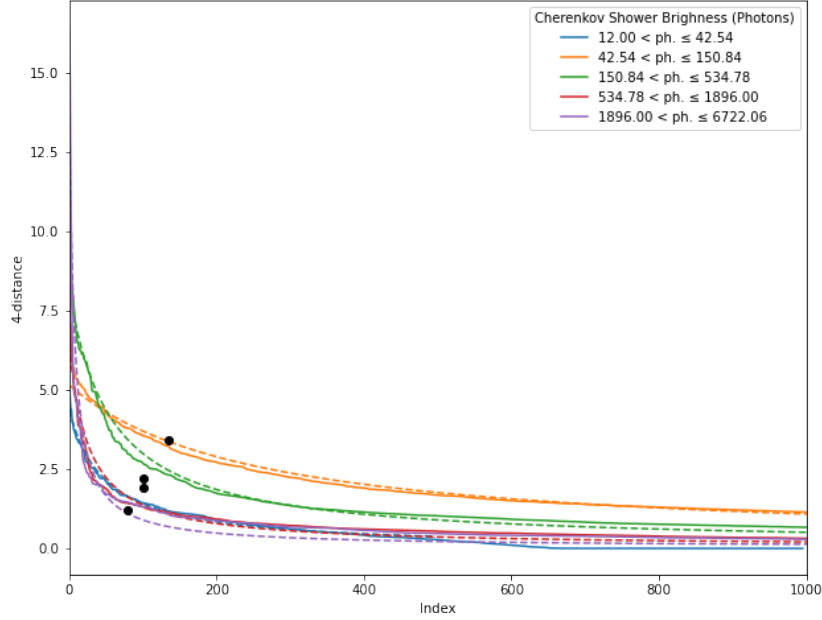


Figure 4.5: Knee plot for gamma ray showers. Each curve corresponds to the accumulated data from many showers within the same brightness range. The dashed lines correspond to a fit by a rational function Eq.(4.2.4). In black are the knee points for proton cosmic ray showers.

parameters. We compare the results yielded by both approaches, sampling and knee curve, for the two types of showers.

In Fig.4.5, we see the knee plot obtained for gamma showers. The shape of the curves is very similar to the ones obtained with proton showers but knees have different positions. This is to be expected since gamma showers should be less spread out than proton showers.

We can make similar observations by comparing the two-dimensional RMSE plots of both proton and gamma ray showers. A clear distinction could not be made by simply looking at Fig.4.1 and Fig.4.2, so we compute instead the ratio between the proton and gamma ray RMSE values as a function of *eps* and *min_samples*. These plots are shown in Fig.4.6. We see the same trend as in Fig.4.5: for the most part of the parameter space, the ratio between RMSE values of proton and gamma ray showers is close to one, which means they are compatible. However, it significantly diverges from one in the minimal regions, which is compatible with Fig.4.5.

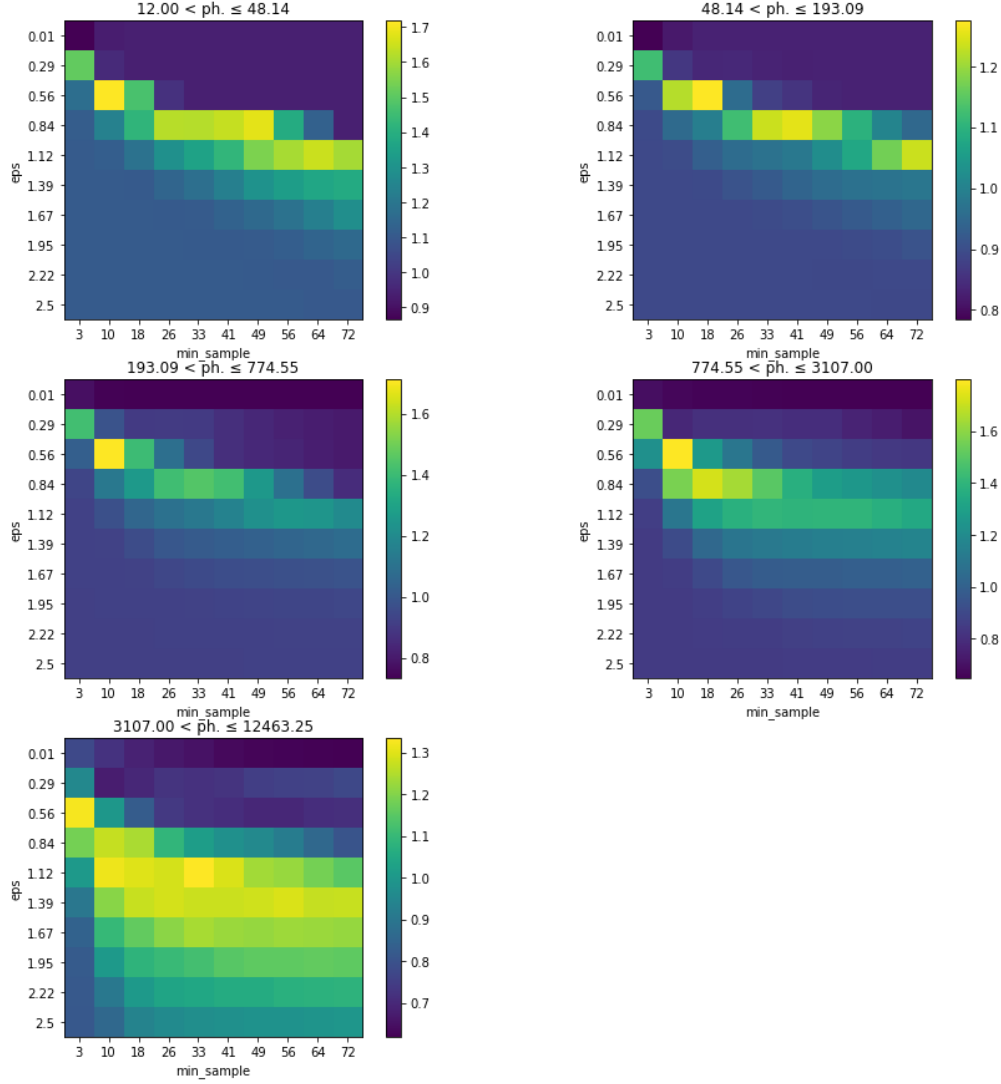


Figure 4.6: Two-dimensional plot of the ratio between the RMSE values for proton and gamma ray showers (colormap), as a function of the eps and min_samples DBSCAN parameters. Each plot corresponds to the average of the results obtained for multiple showers within a different range of brightness (in number of photons).

Bibliography

- [1] B.S. Acharya et al. “Introducing the CTA concept”. In: *Astroparticle Physics* 43 (2013). Seeing the High-Energy Universe with the Cherenkov Telescope Array - The Science Explored with the CTA, pp. 3–18. ISSN: 0927-6505. DOI: <https://doi.org/10.1016/j.astropartphys.2013.01.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0927650513000169>.
- [2] Martin Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.
- [3] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.