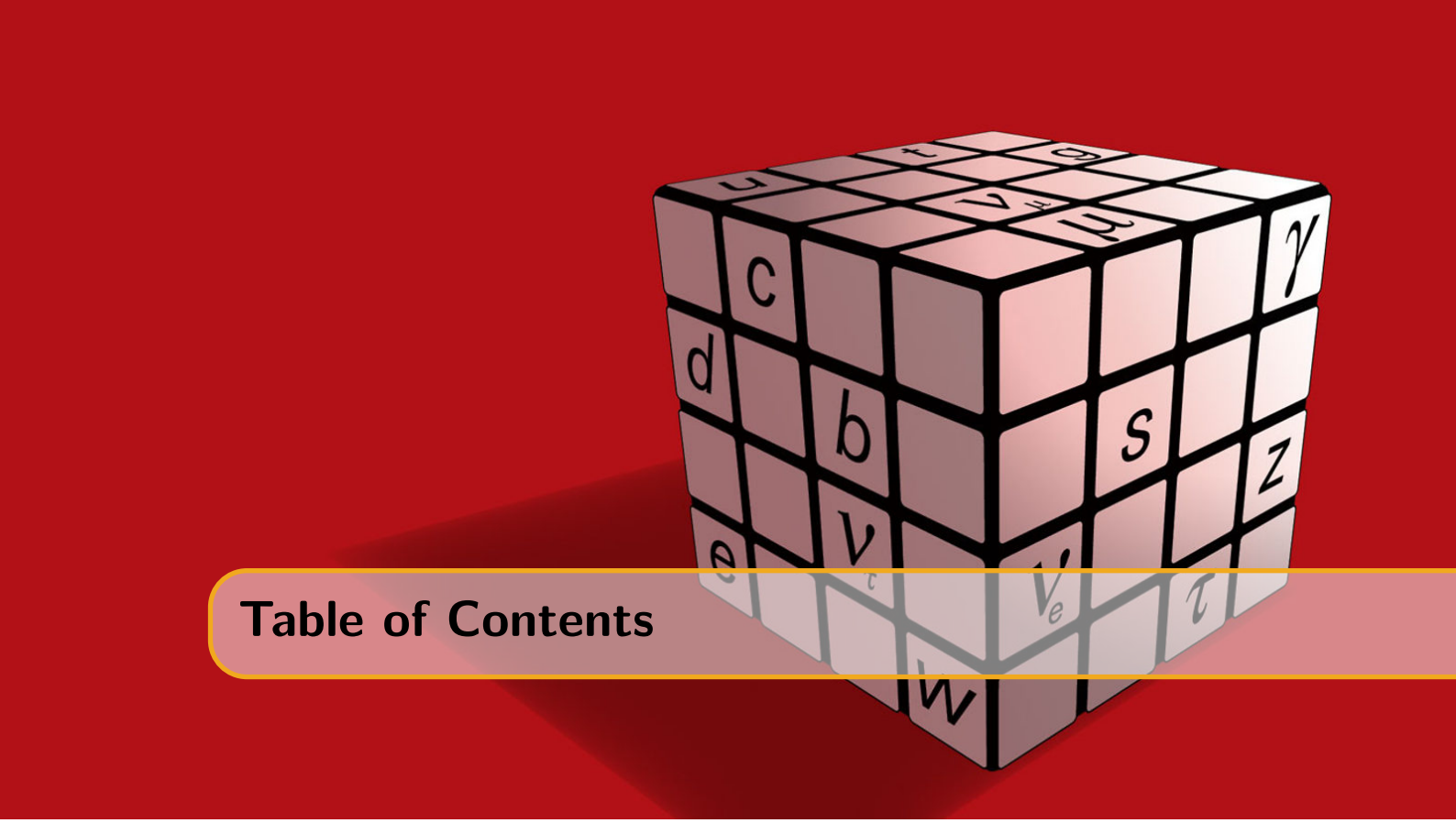


Table of Contents

I Theory and Experiment

II Simulation, Processing and Analysis

1. Reconstruction, Cleaning and Analysis Techniques

Shall I refuse my dinner because I do not fully understand the process of digestion? ~ Oliver Heaviside

Because the in-ice IceCube detector is sparsely distributed, it is not straightforward to unambiguously reconstruct the particle (interactions). The scattering and absorption of photons, tilt of ice sheets, bubble column, etc. lead to uncertainties and make reconstruction challenging. Over the years, multiple reconstruction methods have been developed in the collaboration. They range from very fast (and simple) reconstructions, necessary for online filtering, to slow (and more refined) ones. Multiple reconstruction algorithms have been used in this analysis and are explained in more detail in this chapter.

1.1 Reconstruction

1.1.1 Likelihood

Reconstruction algorithms usually have no unique solutions to describe the set of measured values of an event. The likelihood $\mathcal{L}(\vec{x}|\vec{a})$ describes the probability of a set of parameters $\vec{a}$ to be expressed in a set of experimentally measured values $\vec{x}$. The parameters, $\vec{a}$, typically define the particle's characteristics (energy, direction, position, type, etc.) while the measured values $\vec{x}$ are determined from the detector response (number of PE, timing, position of hit DOMs, etc.). This likelihood is equal to the cumulative probability

$$\mathcal{L}(\vec{x}|\vec{a}) = \prod_i p(x_i|\vec{a}), \tag{1.1}$$

where $p(x, \vec{a})$ is the probability that we measure a certain value x from a set of independent values $\vec{x}$ given an initial set of parameters $\vec{a}$. The best possible guess for the unknown parameters $\vec{a}$ is the most likely set that will result into the experimental values. This is done by maximizing the likelihood $\mathcal{L}$. The reconstruction algorithms below rely on analyzing parameters that assume a single, long track

$$\vec{a} = (\vec{r_0}, t_0, \vec{p}, E_0), \tag{1.2}$$

where $\vec{r_0}$ is the position vector of the particle at a time t_0 with a direction $\vec{p}$ and initial energy E_0.

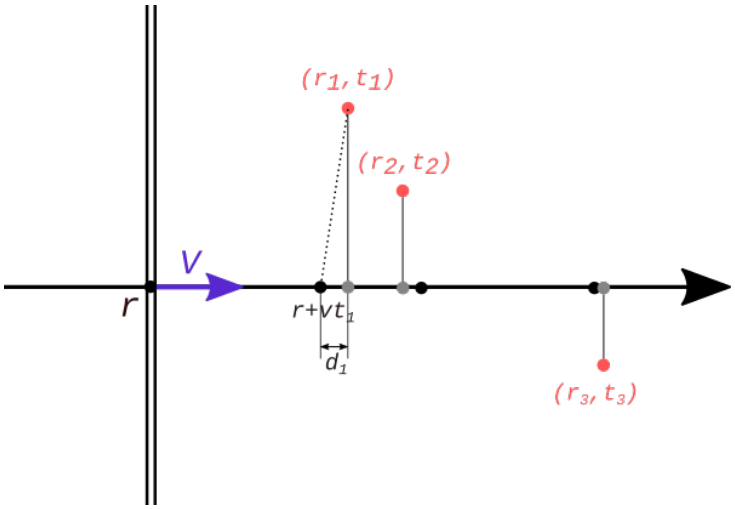


Figure 1.1: Figure illustrating how `LineFit` works. The position, $\vec{r}$, and velocity, $\vec{v}_{\text{part}}$ minimizing the distance of the DOMs to the track is calculated. The dotted line is one of the distances that is minimized in Eq. 1.4.

1.1.2 `Line-Fit`

One of the most simple approaches in constructing a parameter profile is by calculating the track that, overall, has the closest approach of all the hit optical modules and is called `Line-Fit` (LF) [**Ahrens:2003fg**]. If we assume that a particle starts at a position $\vec{r}$ at a time 0 and travels at a velocity of $\vec{v}_{\text{part}}$, then its position at any given time is

$$\vec{r'} = \vec{r} + \vec{v}_{\text{part}} t. \tag{1.3}$$

We want to calculate the best possible estimate of the velocity $\vec{v}_{\text{part}}$ and an initial position $\vec{r}$. Each DOM has a known location, $\vec{r}_i$, and measured time of a pulse, t_i. In this algorithm one assumes that a wavefront perpendicular to the particle's direction is traveling along with the particle. If the velocity $\vec{v}_{\text{part}}$ is fixed, then the position of the particle at later times is known (black points in Fig. 1.1). However, the Cherenkov wavefront should be set at an angle and because scattering, PMT jitter, noise, etc. are not taken into account, this will not agree with the DOM position projected along the particle path (grey dots). The unknown velocity $\vec{v}_{\text{part}}$ and position $\vec{r}$ are the analytical solutions after minimizing the distances d_i as shown in the figure*

$$\begin{aligned} S(\vec{r}, \vec{v}_{\text{part}}) &\equiv \sum_{i=1}^{N_{\text{hit}}} \rho(\vec{r}, \vec{v}_{\text{part}}, \vec{r}_i, t_i)^2 \\ &= \sum_{i=1}^{N_{\text{hit}}} \left(\vec{r}_i - \vec{r} - \vec{v}_{\text{part}} t_i\right)^2, \end{aligned} \tag{1.4}$$

where N_{hit} are the number of pulse hits. The analytical solution by minimizing this equation is equal to

$$\vec{r} = \langle \vec{r}_i \rangle - \vec{v}_{\text{part}} \langle t_i \rangle \quad \text{and} \quad \vec{v}_{\text{part}} = \frac{\langle \vec{r}_i t_i \rangle - \langle \vec{r}_i \rangle \langle t_i \rangle}{\langle t_i^2 \rangle - \langle t_i \rangle^2}, \tag{1.5}$$

where $\langle x \rangle$ denotes the average of a parameter x over all hits i.

Because this is an analytical equation, this algorithm is very fast and therefore often used in online processing.

*Minimizing $r_i - r'$ (dotted line in Fig. 1.1) is the same as minimizing d.

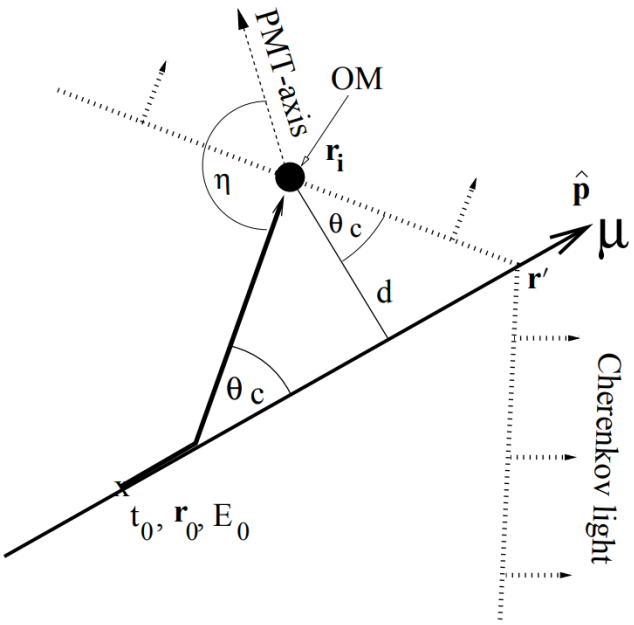


Figure 1.2: Figure illustrating a muon track passing close by an optical module and defining the parameters used in the reconstruction algorithms. Illustration from Ref. [**Ahrens:2003fg**].

1.1.2.1 `Improved Line-Fit`

While disregarding the Cherenkov profile is inherent to the simplified LF model for computational reasons, removing hits generated by photons that scattered for a significant length of time will mitigate the effect of ignoring the photon scattering in the ice. It was found that a basic filter could identify these scattered hits, and improve accuracy by almost a factor of two by removing them from the dataset. More formally, for each hit h_i, the algorithm looks at all neighboring hits within a neighborhood of μ, and if there exists a neighboring hit h_j with a time stamp that is t earlier than h_i, then h_i is considered a scattered hit, and is not used in the basic reconstruction algorithm. Optimal values of μ and t were found to be 156 m and 778 ns by tuning them on simulated muon data [**Aartsen:2013bfa**].

This "delay cleaning" is done by computing the Huberfit on the remaining data points and minimizing

$$\sum_{i=1}^{N_{\text{hit}}} \phi(\rho(\vec{r}, \vec{v}_{\text{part}}, \vec{r}_i, t_i)), \tag{1.6}$$

where ρ is defined in Eq. 1.4 and the Huber penalty function ϕ is defined as

$$\phi(\rho) \equiv \begin{cases} \rho^2 & \text{if } \rho < \mu \\ \mu\,(2\rho - \mu) & \text{if } \rho \geq \mu \end{cases} . \tag{1.7}$$

Because of the overal performance increase of this method, all LF computations were done with the improved version (although still often referred to as "Line-Fit").

1.1.3 SPE and MPE

A more intricate method of track reconstruction is done by taking the geometrical shape of the Cherenkov cone into account and relying on simulation fits where a seed track is implemented (usually from the fast `Line-Fit` algorithm).

Let us assume a particle is traveling close to a DOM with parameters defined in Eq. 1.2 as illustrated in Fig. 1.2. The minimal distance of the track to the DOM is equal to d and the PMT-axis (downwards relative to DOM) has an angle offset of η degrees of the Cherenkov wave

direction. In perfect conditions, the *time residual* (time between the observed hit time and the "expected" time) is a delta function centered around 0, where

$$t_{\text{res}} \equiv t_{\text{hit}} - t_{\text{geo}}, \tag{1.8}$$

with

$$t_{\text{geo}} = t_0 + \frac{\vec{p}\cdot(\vec{r}_i - \vec{r}_0) + d\cdot\tan(\theta_c)}{c_{\text{vac}}}, \tag{1.9}$$

which is equal to the time of the particle to travel from the position $\vec{r}_0$ to $\vec{r}'$ as illustrated in the figure. The accompanying Cherenkov wavefront that sent out photons at a time t_0 from $\vec{r}_0$ will cross the DOM when the particle is at a position $\vec{r}'$. Due to noise effects, PMT jitter, light from secondary interactions, DOM orientation, etc. the time risidual is smeared and shifted. The p.d.f. was estimated with photon simulations in ice and fitted to a Pandel function [**Ahrens:2003fg**]. The time likelihood profile for single photons i at the locations of the hit DOMs is then

$$\mathcal{L}_{\text{time}} = \sum_{i=1}^{N_{\text{hit}}} p_1(t_{\text{res}}|\vec{a} = d_i, \eta_i, ...). \tag{1.10}$$

An initial particle position and direction are found by maximizing the likelihood and iterated a couple of times to find the global maximum instead of a local. This fitting is called the Single PhotoElectron (SPE) fit.

The description of single photons arriving at the optical modules cannot be correct since electrical and optical signal channels can only resolve multiple photons separated by a few 100 ns and $\approx$ 10 ns, respectively. In the Multi-PhotoElectron (MPE) fit, one accounts for the fact that the early photons in a DOM hit scattered less in the ice. The p.d.f. for the first photon out of a total of N to arrive with a time residual of t_{res} is

$$p_N^1(t_{\text{res}}) = N\cdot p_1(t_{\text{res}})\cdot\left(\int_{t_{\text{res}}}^{\infty} p_1(t)dt\right)^{(N-1)} = N\cdot p_1(t_{\text{res}})\cdot(1 - P_1(t_{\text{res}}))^{(N-1)}, \tag{1.11}$$

where P_1 is the cumulative distribution of the single photon p.d.f..

1.1.4 `Millipede`

?? To have a better handle on the particle energy and cascades along the track, the module `Millipede` was developed. The number of photons seen at each optical module depends on multiple factors that were mentioned throughout this text, such as the ice characteristics, timing, etc. In this module, the expected number of photons is said to depend on the energy that was deposited along a track and a *light yield factor* that depends on the DOM position and the location of emission

$$\begin{aligned} N_{\text{exp},k} &= \rho_k + \sum_{i=1}^{n} \Lambda(\vec{r}_k, \vec{r}_i')E_i \\ &= \rho_k + \vec{\Lambda}(\vec{r}_k, \vec{r}_i')\cdot\vec{E}, \end{aligned} \tag{1.12}$$

where k refers to a certain DOM and i refers to a certain energy deposit such as illustrated in Fig. 1.3.

The likelihood is assumed to follow a Poisson distribution with a mean equal to the expected amount of photons, $N_{\text{exp},k}$

$$\mathcal{L}_k = \frac{\left(\vec{\Lambda}\cdot\vec{E} + \rho_k\right)^{N_{\text{seen},k}}}{N_{\text{seen},k}!} e^{-\vec{\Lambda}\cdot\vec{E} - \rho_k}. \tag{1.13}$$

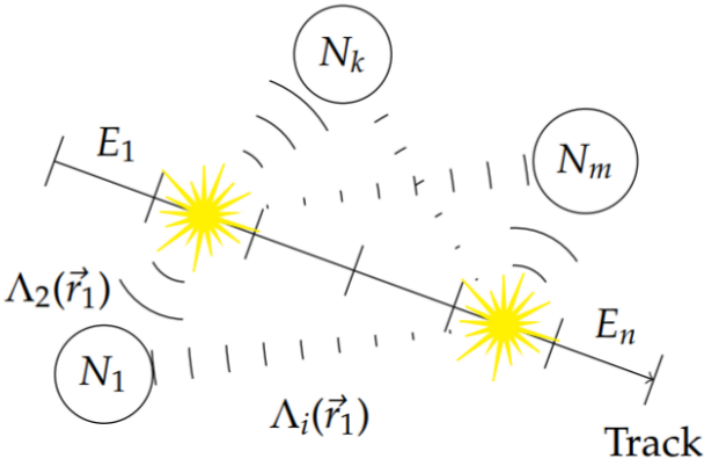


Figure 1.3: Illustration of the working principles of the `Millipede` toolkit. A track is subdivided into segments that each deposit a certain energy, E_i. Different segments can contribute to the total number photons seen per DOM, N_k.

For easier, faster and more accurate computation the logarithm of the likelihood is used

$$\begin{aligned}\ln \mathcal{L}_k &= N_{\text{seen},k} \ln\left(\rho_k + \sum_{i=1}^{n} \Lambda(\vec{r}_k, \vec{r}_i')E_i\right) - \ln\left(N_{\text{seen},k}!\right) - \sum_{i=1}^{n} \Lambda(\vec{r}_k, \vec{r}_i')E_i - \rho_k \\ &= N_{\text{seen},k} \ln\left(\rho_k + \vec{\Lambda}(\vec{r}_k)\cdot\vec{E}\right) - \vec{\Lambda}(\vec{r}_k)\cdot\vec{E} - \rho_k - \ln\left(N_{\text{seen},k}!\right)\end{aligned} \tag{1.14}$$

Maximizing the total likelihood (summing over all m DOMs) with respect to the energy gives

$$\nabla_{\vec{E}} \ln \mathcal{L} = \nabla_{\vec{E}} \sum_{k=1}^{m} \ln \mathcal{L}_k = \sum_{k=1}^{m} \left(\frac{N_{\text{seen},k}\vec{\Lambda}(\vec{r}_k)}{\vec{\Lambda}(\vec{r}_k)\cdot\vec{E} + \rho_k} - \vec{\Lambda}(\vec{r}_k)\right) = 0. \tag{1.15}$$

This equation holds if all terms in the sum vanish, i.e. if for all DOMs holds that

$$\begin{aligned}N_{\text{seen},k} &= \vec{\Lambda}(\vec{r}_k)\cdot\vec{E} + \rho_k \\ &\overset{\text{Eq.1.12}}{=} N_{\text{exp},k}.\end{aligned} \tag{1.16}$$

This can be written in a set of linear equations

$$\vec{N} - \vec{\rho} = \mathbf{\Lambda}\vec{E}, \tag{1.17}$$

where

$$\mathbf{\Lambda} = \begin{pmatrix} \Lambda(\vec{r}_1, \vec{r}_1') & \cdots & \Lambda(\vec{r}_1, \vec{r}_n') \\ \vdots & \ddots & \cdots \\ \Lambda(\vec{r}_m, \vec{r}_1') & \cdots & \Lambda(\vec{r}_m, \vec{r}_n') \end{pmatrix}, \tag{1.18}$$

is the *response matrix* and has to be inverted to find the energies in the vector $\vec{E}$. It describes the DOM response to light output from certain segments along a track. The entries in this matrix come from simulations that produce spline tables. Simplified sources, such as minimum ionizing muons and isotropically emitting point sources are simulated in Monte Carlo simulations at certain discrete points. Interpolation is done using spline functions. More information, such as how timing information can be implemented, can be found in Refs. [**millipedeinternal**, **stefthesis**].

1.1.5 `FiniteReco`

`FiniteReco` is a module that tries to reconstruct if particles are starting, stopping, contained or through-going. The hit DOMs around a seed track are checked to have seen light and the first and last emission points along the track are used to check the possible hypotheses.

Because the edges of the detector are not well defined*, the likelihoods of individual DOMs to have seen a hit lead to a total likelihood that doesn't give a conclusive answer, but the starting and stopping probabilities can be compared to a through-going track hypothesis.

1.1.6 `Paraboloid`

In Sections 1.1.2 and 1.1.3, we discussed how a particle's direction could be estimated. The `Paraboloid` module tries to provide an estimate for the error on this direction. A highly energetic muon with hunderds of hit DOMs will lead to a much better directional resolution than a dim track where only a handful of DOMs are hit. In general, the likelihood space around the estimated direction is scanned and compared to the likelihood of the initial track estimation. This method also gives a robust estimation if the initial track direction is in fact located at the global maximum likelihood or a local one. `Paraboloid` constructs a grid of zenith and azimuth points near the minimum and for each point on the grid it does a three-parameter minimization for the vertex holding the zenith and azimuth constant. The likelihood values for each point on the grid are then fit to paraboloid using a χ^2 minimization since the shape of the log-likelihood space near the minimum should have a paraboloid (2D parabola) shape. Of importance are the parameters of the corresponding error, which is assumed to correspond to an ellipse for the 1σ contours.

The module computes the lengths of the semimajor and semiminor axes of the 1σ error ellipse σ_1 and σ_2†. It was found that the quadratic mean of both uncertanties provides for a good single-valued estimate for the angle uncertainty

$$\sigma_{\text{para}} = \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}}. \tag{1.19}$$

Since σ_1 and σ_2 should follow univariate Gaussian distributions, σ_{para} will be the radius parameter in a bivariate distribution, meaning that the mean should be set to 1.177σ‡. This means, that if we calculate the great circle distance between the MC truth of the signal particle with the reconstruction direction and divide it with σ_{para}, the distribution should peak at 1.177 (mean). This variable is called the *paraboloid pull* and should be compared to an energy related variable (here the number of hit DOMs (NCh) is used). In Fig. 1.4, it is clear that there is an offset, which is seen in all analyses, that can be explained by multiple factors in Monte Carlo simulations but mainly stems from our incomplete knowledge and non-perfect simulations of the ice.

More information can be found in Ref. [**Neunhoffer:2004ha**].

1.2 Pulse cleaning

As explained in Section **??**, each DOM in IceCube has an intrinsic noise rate. This dark noise is observed in every triggered event and seen as random hits in the detector added to the hit pattern of tracks and cascades. These spurious hits are a large nuisance factor in event reconstructions, leading to misidentification and errors in the result. Noise cleaning should be done in early stages of event processing and analysis to reduce a large rate of bad reconstructed events that pass cut selections. One of the most conservative ways is to only look at HLC hits (*HLC cleaning*), but is too demanding for most low-energetic events that will have multiple hits from isolated DOMs.

*Imagine a cascade 20 m below the lowest DOMs. It is still possible for light to reach the bottom modules of the detector.

†The confidence intervals σ_θ and σ_ϕ can be found by rotating the minor and minor axes σ_1 and σ_2. How this is done can be found in the literature, but is of no importance here.

‡The CDF of the Rayleigh distribution $1 - e^{-x^2/2\sigma^2}$ is equal to the containment for a bivariate normal distribution. Implementing $x = 1.177\sigma$ yields a factor of 0.5.

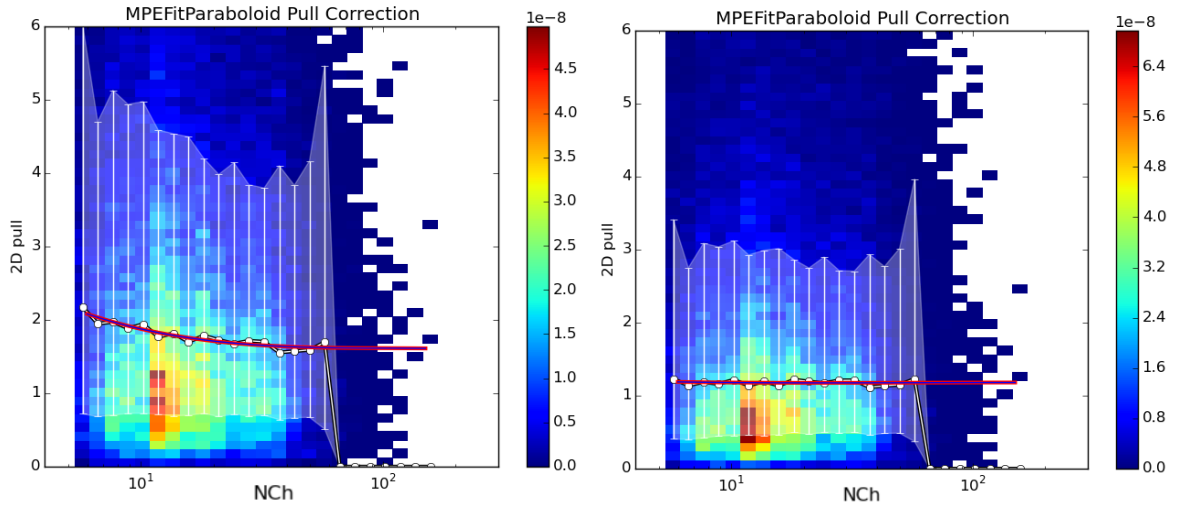


Figure 1.4: *Left:* Paraboloid pull in function of the number of hit DOMs shows that the global average (red/purple line) is not centered at 1.177. *Right:* Pull after correction.

Another, more conservative method, is the `seededRT` algorithm. This method relies on the "*RT*-cut", which was already implemented in the time of AMANDA operations. R is a designed radius and T refers to the time between multiple possible pulse times (e.g. the pulse of one DOM starts during the time window of a second DOM's pulse, or stops during the time window). The full description can be found in Ref. [**RTcutwiki**], but can be summarized as follows: DOMs are required to be in a temporal and spacial coincidence that is physically possible (e.g. signal between DOMs cannot exceed the speed of light in vacuum). This method is however computationally expensive since all DOM pairs have to be looped over*. The `seededRT` algorithm takes a subset of seeds that are considered to be mostly signal related hits. These seeds can be provided by, for example, using HLC information. By adding all further hits found within the seed's *RT*-range to the list of seed hits and iterating until a convergence, only those (SLC) hits are kept that cluster around the initial seed hits. Outlying noise hits are supposedly not added and thus removed in the cleaned output. This method does not scale as drastically as the original *RT*-cut method.

1.3 `IceHive`

In Section **??**, it was explained how multiple triggers were combined into one global trigger. In a first step, Q-frames are simply re-split into the individual events that belong to the different subtriggers. In about 15% of cases the data read out in one of these P-frames contains more than one primary interaction. This pile-up effect is referred to as *coincident events*. It is a direct result of the traversal time of a couple of microseconds in the detector†, the large flux of low-energetic events and the trigger time windows of a couple of microseconds. This can be problematic for reconstructions, as can be seen in Fig. 1.5 where two downgoing muons can be reconstructed as an upgoing track.

There are two modules that try to clean events more thoroughly than pulse cleaning alone. The first is `TopolocalSplitter` (TS), which starts from the Q-frames and loops over pulses and splits the event into clusters of pulses that contain at least a number of causally‡ connected pulses within a certain time window. Some extra cleaning, similar to `seededRT` cleaning, is done in addition and can split coincident events that have overlapping readout windows, but are geometrically separated.

*The number of pairs for n DOMs is equal to $\frac{1}{2}n(n-1)$ and scales with n^2.

†The speed of light in vacuum is equal to ≈ 0.3 m/ns, meaning the particle travels around 100 m in 0.3 μs, without accounting for the delayed photon propagation necessary for detection.

‡The time between two DOM hits cannot be less than the time that light may have taken.

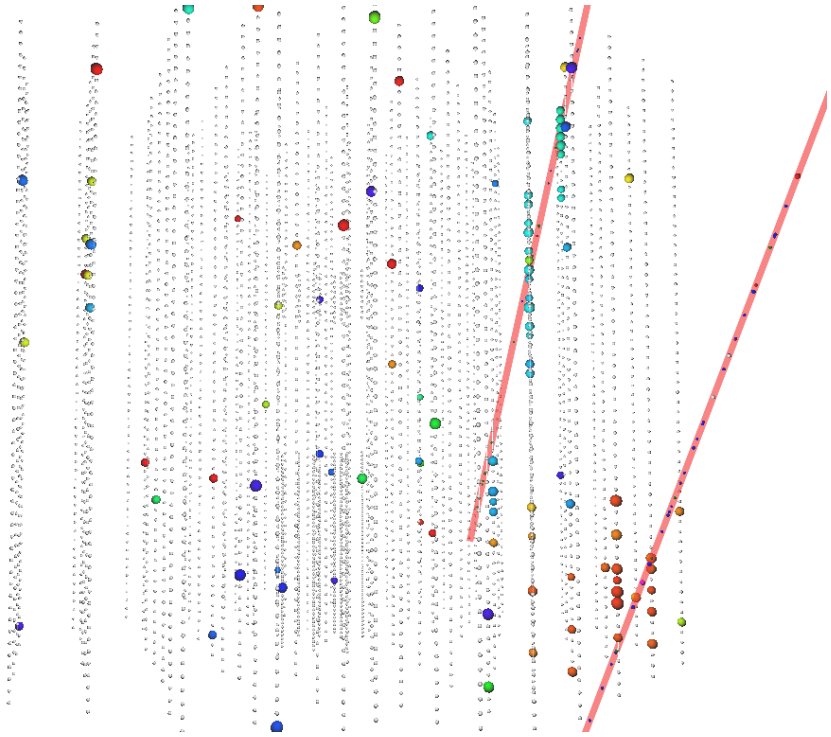


Figure 1.5: Event display of a simulated coincident event of two downgoing muons. The colors of the event range from red (early) to blue (late). The first muon hits the bottom of the detector, while the second traverses mainly the upper part. These events are often reconstructed as a single up-going event and therefore result in a large background contribution. The scattered isolated hits are due to noise effects and mostly removed by pulse cleaning.

The second module, and also used in this analysis, is called `IceHive`. A full description can be found in the doctoral thesis of M. Zoll [**mzollthesis**]. The module consists of two main parts: one that splits events and handles coincident events, `HiveSplitter`, and another that has a refined pulse cleaning, `HiveCleaning`.

1.3.1 `HiveSplitter`

The module assumes that individual particles will create *clusters* of hits in the detector. A cluster can grow within a certain time window, but is separated from another cluster if it's not spatially connected. An initial cluster is formed if the multiplicity of hits exceeds a certain threshold (usually 3 or 4). The main difference in this module versus `TopologicalSplitter` is that it uses hexagons to describe the detector instead of assuming a spherical parameterization. It makes more sense to optimize the search volume, where hits are clustered together, with a shape that describes the detector well and uses a discrete spacing between larger volumes instead of a uniformly growing sphere. The hexagonal shape is set by defining three heights. The first height is defined along the string of the hit DOM and is equal to the vertical distance along the string. The second height is the vertical height along the neighboring strings. The third hight is the vertical height along the next-to-neighboring strings. An example is shown in Fig. 1.6.

When the active region is set, there is an additional check to see if DOMs can be "connected". `IceHive` assumes certain emission profiles (for both cascades as tracks) where light is produced. Three possible connections are assumed:

1. Hits occur at the same time, but at a spatial distance in agreement with the Cherenkov emission profile (hits C&1 and 2&3 in Fig. 1.6).
2. Hits occur at a different time and a different location, but in agreement with the Cherenkov emission profile (hits C&2 in Fig. 1.6).
3. Hits on topologically identical sites of an emission pattern that has moved along with the propagation of the particle (hits C&3 and hits 1&2 on Fig. 1.6).

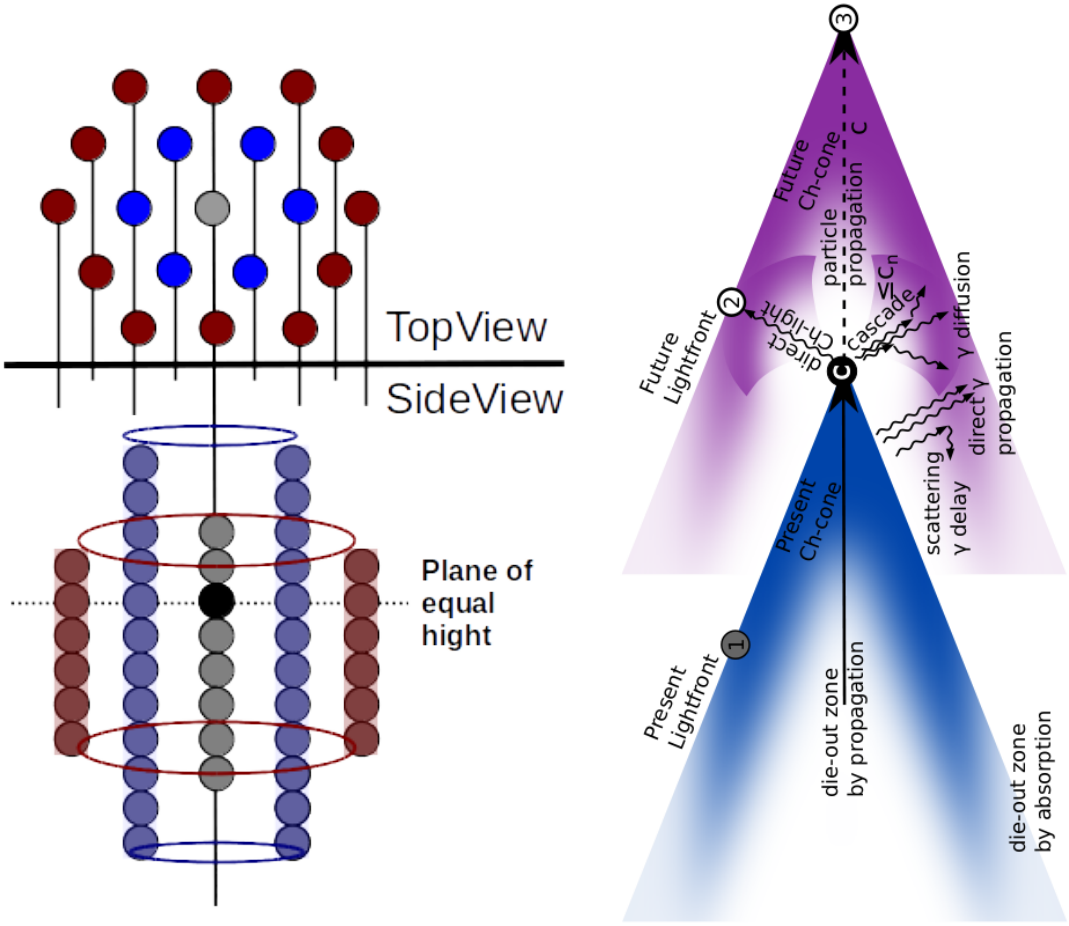


Figure 1.6: *Left:* The black circle illustrates a DOM that triggered a hit in the detector. The grey circles symbolize the DOMs along the string of the hit DOM. The number of DOMs that can be included in the active volume depends on the height defined by the module. The blue/purple DOMs belong to the neighboring strings and the red/brown DOMs to the next-to-neighboring strings. The heights of both these sets of DOMs are also set by the module. This example shows $h_2 > h_1 > h_3$, the heights are also asymmetric in this example. *Right:* Illustration of Cherenkov emission profile of a traversing particle. Both figures from Ref. [**mzollthesis**].

These clusters are finally separated into different P-frames and thus regarded as distinct events.

1.3.2 `HiveCleaning`

Additionally, a similar cleaning as explained in Section 1.2 can be performed. Isolated hits that do not have neighboring hits occurring within a certain distance and time window, are removed. The main difference between this and `seededRT` cleaning is that the module again uses the hexagons as defined in the previous section.

1.3.3 Remark

The usage of `IceHive` has a great performance in separating coincident events, but often "overperforms" and splits clusters of hits that are originating from the same particle. This is predominantly the case for dim tracks that have large separations in between clusters (most of the triggered SMP events are of this type). It is because of this that the module `CoincSuite` was designed.

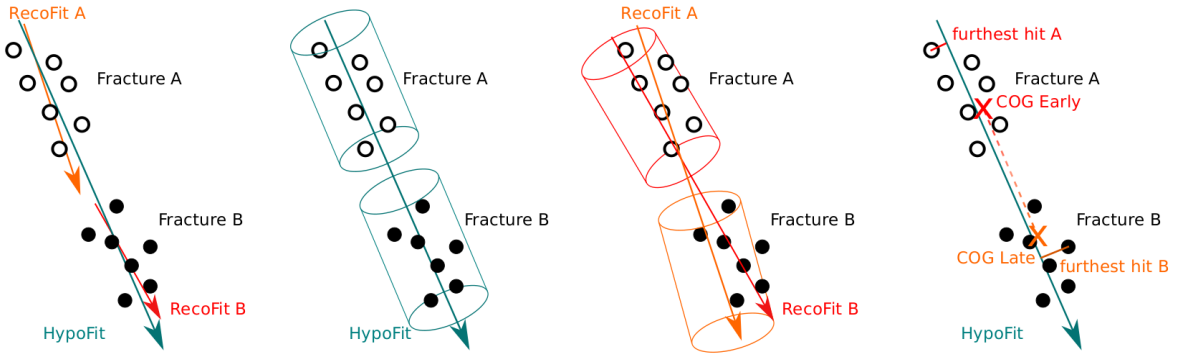


Figure 1.7: Schematic illustrations of possible recombination scenarios with the `CoincSuite` module. Two events are compared to each other or the combined event (HypoFit) in several different ways.

1.4 CoincSuite

Several testing algorithms allow to check if two or more split P-frames can originate from a single event. Five different scenarios were tested in this analysis, the first four are also chronologically shown in Fig. 1.7:

1. Cluster alignment: the reconstructed direction of the individual clusters is compared to the direction of a reconstruction that uses the combined hits (HypoFit). The directions should be within a certain criticle angle.
2. Cylinder cluster containment: the DOMs of the individual clusters should be able to be grouped together in a cylinder that has its center and direction along the HypoFit.
3. Cylinder cluster alignment: a cylinder around the reconstruction of each cluster is draw. The cylinders should overlap within a certain fraction.
4. COG* connection: the second quarter of the COG of the first cluster and the third quarter of the COG of the second cluster are computed. These COG should lie close enough and have to be in the vicinity of the HypoFit.
5. Velocity test: tests if the velocity of the HypoFit is close to the speed of light.

The combination of `IceHive`, which does a very good job in cleaning events, but often overperformes and splits events that shouldn't be, and `CoincSuite`, which recombines events that were wrongfully split, leads to a very powerful tool to clean events.

1.5 Analysis techniques

Komt hier iets dat je gebruikt in het volgende hoofdstuk maar nog niet hebt uitgelegd?

1.6 Boosted Decision Tree classifiers

Given a certain event with a fixed set of variables that are constructed in an analysis, the question remains if the event is in fact a *signal* or *background* event. One can rely on Monte Carlo simulations to get a handle on the variable distribution in both sets. The most general and still widely used method is to use a cut-and-count approach where a cut is placed on a certain variable that discards events that fail the requirement. A Boosted Decision Tree (BDT) inspects a set of set variables and classifies an event with a score that ranges from -1 to 1. The higher the score, the more an event is regarded as signal-like. How this is done is given in more detail below. Boosted decision trees rely on multiple individual trees. Therefore, we will first explain how a single tree classification works.

*Similar to COM (Center Of Mass), the COG is a weighted average position of the hit DOMs. DOMs that register more light get a heigher weight.

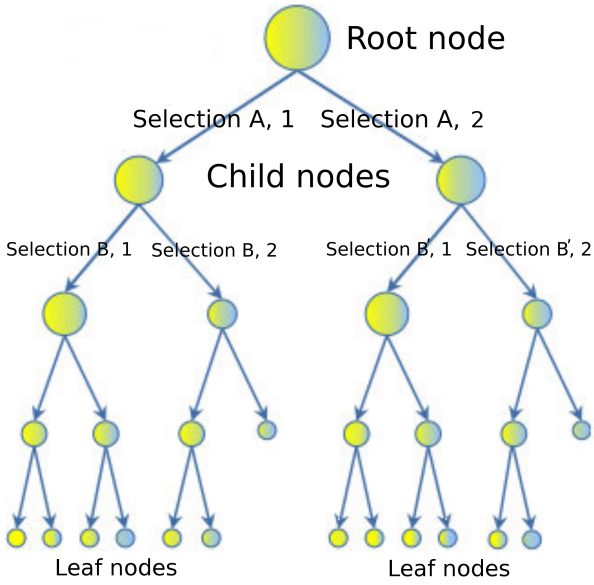


Figure 1.8: Illustration of decision tree scheme. Selection B and B' denote a selection on the same variable, but other requirement.

1.6.1 Stucture

The goal of a decision tree is to determine if an event is signal- or background like. It uses a tree-like structure where certain selection criteria are used at different nodes as illustrated in Fig. 1.8.

A decision tree is a binary tree that places an event into a certain node depending on the selection at a node. The depth of a decision tree can be arbitrarily long, but is determined by a set of criteria defined in Section 1.6.2. An event consists of a certain set of variables $X = x_1, x_2, ...x_n$ that are used in the classification. Before any selection criteria, the event is said to be represented by the *root node*. A binary selection then determines to which *child node* the event should be classified, for example:

$$\begin{aligned} \text{Selection A} &= x_1 > y_1 \quad \text{(option 1)} \\ &= x_1 \leq y_1 \quad \text{(option 2)}, \end{aligned} \tag{1.20}$$

where y_1 is the cut value for variable x_1. Similarly, the other selections deterimine where the event is eventually placed. The last nodes are referred to as *leaf nodes* and hold the probabilities of whether an event is more signal- or background-like. These probabilities are translated into a score ranging between -1 (background) and 1 (signal).

1.6.2 Training

To construct a decision tree, one first has to "train" the algorithm. Given a certain "signal set" and "background set", all variables used in the BDT are histogrammed and at each bin for each variable the "best cut" is set at the first node selection. To determine the optimal cut, we first define the purity of a node, p, by

$$p = \frac{\sum_s w_s}{\sum_s w_s + \sum_b w_b}, \tag{1.21}$$

where w_s and w_b refer to the weights of the signal and background events and the Gini index, g,

$$g(p) = p(1-p), \tag{1.22}$$

is used as a separation variable in this work*. Using the Gini index, the separation gain determines the effectiveness of the cut

$$\Delta S = g_p \cdot \sum w_p - \left(g_l \cdot \sum w_l + g_r \cdot \sum w_r \right), \tag{1.23}$$

where g_p and w_p denote the Gini index and weights of the parent nodes and similarly for the left and right child nodes. The cut that gives the highest separation gain is subsequently selected. The algorithm stops when one of the following criteria is met:

- a node only consists of signal or background events;
- a certain predefined maximal depth is reached;
- splitting would cause a child node to have less than a predefined minimal amount of events left;

and therefore determines the size of a tree. These selection criteria are necessary to avoid overtraining (see Section 1.6.4).

1.6.3 Boosting

As already implied in the text above, a BDT consists of a *forest* of decision trees. A user specified number of individual decision trees are trained sequentially, with a boosting process in between each training. Boosting consists of adjusting the weights of individual events according to whether the previously trained tree classifies them correctly. In this work, the AdaBoost† algorithm was used for boosting in which the score of an event is a weighted average of the scores the event receives from each tree in the forest [**FREUND1997119**].

A BDT may informally be called a "boosted decision tree", but it must be understood that there are actually many trees (typically hundreds), and that boosting is a process that occurs between the training of consecutive trees. The approach makes use of the power of numbers: many weak single decision trees combined can be more powerful than one very good decision tree. In general, boosting follows the following steps:

1. Train a weak model on training data.
2. Compute the error of the model on each training example.
3. Give higher importance to examples on which the model made mistakes.
4. Re-train the model using "importance weighted" training examples.
5. Go back to step 2.

An example is given in Appendix E.

If $I(s, y)$ is a function equal to 0 when, after tree classification, the sample test score, s, is equal to its true identity, y, then the error rate for a tree is equal to

$$\epsilon = \frac{\sum_i w_i I(s, y)}{\sum_i w_i}. \tag{1.24}$$

The boosting factor for the tree is defined as

$$\alpha = \beta \cdot \ln\left(\frac{1-\epsilon}{\epsilon} \right), \tag{1.25}$$

with β a user defined *boosting beta* and changes the weight of the tree to

$$\begin{array}{cc} w' = w \cdot \exp(\alpha), & w' = w \cdot \exp(-\alpha) \\ \text{(correct classification)} & \text{(incorrect classification)}, \end{array} \tag{1.26}$$

*Other possible separation variables include the cross entropy $-p \cdot \ln(p) - (1-p) \cdot \ln(1-p)$ or the misclassification error $1 - \max(p, 1-p)$

†Short for Adaptive Boosting.

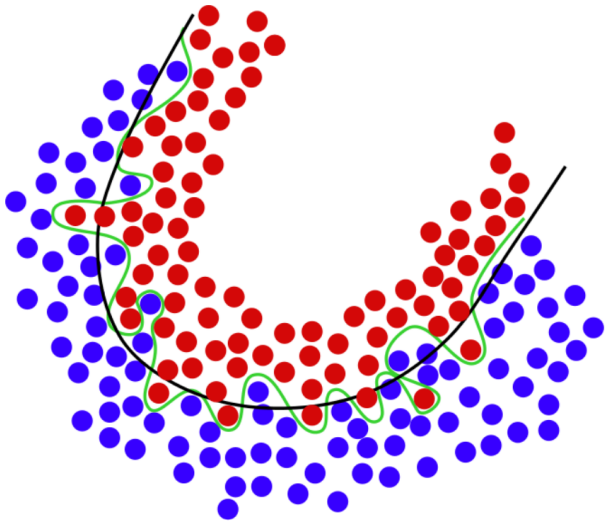


Figure 1.9: Example of overtraining. In black, the theoretical line between background (blue points) and signal (red points) is given. An overtrained BDT will have a (perfect) selection such as the green line. Illustration from [**boserpdf**].

after which the weights are renormalized so that $\sum w' = 1$. The process is repeated until the number of predefined trees is reached.

Due to its definition, the boost factor α will give good classifiers (which have low error rates) large boost factors. Events that are misclassified are then given larger weights, making the algorithm more likely to classify them correctly in the subsequent tree classifier.

Once the entire BDT is trained, the events can be given a score based on the multiple tree classifiers. This is done by taking the weighted average of all the scores in the individual tree classifiers, using its boost factor α as the weight of the tree. The score of an event i is then given by

$$s_i = \frac{\sum_m \alpha_m \cdot s_{i,m}}{\sum \alpha_m}, \tag{1.27}$$

where we loop over the individual trees denoted with index m.

1.6.4 Overtraining

BDTs are very powerful tools, but if not used correctly could lead to problems that are not easy to spot at first sight. Assume we train our BDT with a certain signal set and background set. If the BDT is trained up to the point of classifying statistical fluctuations, there is said to be *training sample overtraining*. An illustrative example is given in Fig. 1.9. Another example is data/MC overtraining. The former can be dealt with with the use of *pruning*, while the latter should show clear data/MC agreement.

1.6.4.1 Pruning

The problem with overtraining is essentially that there are certain splits in a classifier tree that are too specific and less important. In the method of *cost complexity pruning*, for each node the complexity is calculated as

$$\rho = \frac{\Delta S}{n_{\text{leaves}} - 1}, \tag{1.28}$$

with ΔS the separation gain as defined in Eq. 1.23. The subtree of the node with the smallest value of ρ is removed and this is done repeatedly until a desired *pruning strength** is reached.

*A parameter on a scale from 0 to 100, which specifies the percentage of the pruning sequence to actually execute. 0 means no pruning is done and 100 signifies only a single root node remaining.

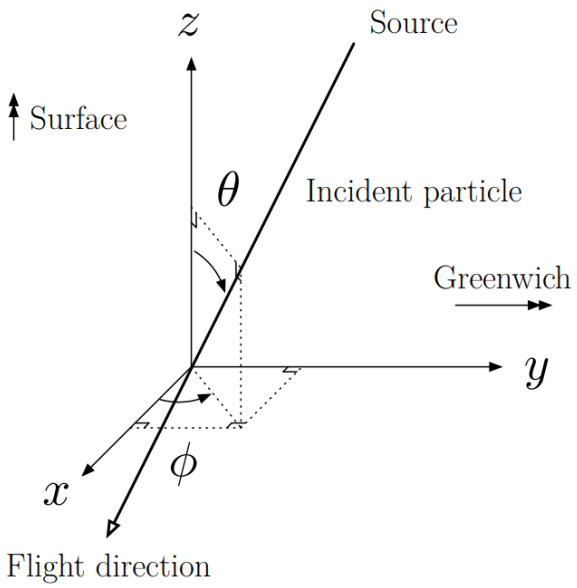


Figure 1.10: The IceCube Coordinate system.

BDTs are always trained and tested with different subsets of the same type (background or signal) of event. If the training set has significantly different scores than the testing set, the sample is most probably overtrained and one has to change the BDT input parameters. Typically, one uses Kolmogorov-Smirnov testing to indicate if the BDT score distributions from the training and testing sets could follow the same distribution pattern. As a rule of thumb, a $p_{KS} \lesssim 0.01$ indicates overtraining beyond the level of comfort.

1.6.4.2 Data-MC agreement

Nodig?

1.7 Minimal-Redundancy-Maximum-Relevance

Having multiple variables that are able to discriminate signal from background events is a neccesary tool to ensure to conclude statements about a certain theory or exotic phenomenon. Single variables can show promising results, but when multiple variables are highly correlated much of the discriminative power diminishes. When using BDTs, analyzers often try to include variables and remove them if they show to be highly correlated in a trial-and-error fashion.

In this analysis, I made use of a technique that was originally developed for data in biological sciences but can be used for most analyses that involve "data mining ". Variables from a large sample set were selected with the condition of minimal-Redundancy-Maximal-Relevance (mRMR). To optimize the characterization of a certain class of events with a set of variables, these variables are selected with a *maximal relevance*. "Relevance" is characterized in terms of correlation of mutual information. Because combinations of individually good features do not necessarily lead to good classification performance, there is the additional requirement of *minimal redundancy* [**1453511**].

In this analysis, mRMR was used to rank variables from a large set according to their importance and proved to lead to low correlated variables (see Fig. ???).

1.8 IceCube coordinate system

Lastly, when referring to positions and directions one first has define a coordinate system to be able to uniquely define these variables. The system is shown in Fig. 1.10.

The center of the coordinate system is set close to the geometric center of the detector, at about 2000 m below the surface of the ice. The y-axis of the coordinate system is aligned with the Prime Meridian pointing toward Greenwich (United Kingdom). The x-axis is set perpendicular to the y-axis pointing in a, 90° clockwise direction. The z-axis is set perpendicular to the xy-plane, pointing upwards, normal to the Earth's surface.

A particle's direction is defined with zenith and azimuth angles, θ and ϕ respectively. The zenith angle is measured relative to the positive z-axis and the azimuth angle is measured counterclockwise from the positive xy-plane.

2. The SPACE Analysis

After introducing the detector workings, reconstruction and analysis techniques, background contributions and the signature of the signal, this chapter gives an overview of analysis. Starting from data processed with basic reconstructions and requirements a workflow was set up to try to discriminate events that are most likely of known physical interactions from the rare events that are sought for in this analysis. These events would originate from the theoretical particles with an anomalous charge (see Chapter **??**). The analysis was adopted the "SPACE" analysis, which stands for a "Search for Particle with Anomalous ChargE".

2.1 Filter selection

As explained in Section **??**, the data is processed through multiple filters. Since this analysis is the first of its kind in the collaboration, no processed dataset from other analyses was used. Filters had to be selected for proper comparison of data and Monte Carlo and I have chosen to optimize the signal to background ratio to select which filters should be included. An illustration is given in Fig. 2.1. This filter selection will be referred to as *Level2b*, as a simple addition to filter processing in Level2 (see Section **??**).

2.1.1 VEF

The Vertical Event Filter (VEF) is designed to be used for oscillation and Earth WIMP analyses and makes use of the string trigger (see Section **??**). An SMP that travel alongside a string, or closeby, can trigger optical modules while the total light yield of an event is low, making this filter an ideal addition to the filters that are selected. In addition, the filter removes HLC hits in the top 5 DOM layers to reduce the muonic component from air shower events. Other selection cuts, try to optimize the search efficiency for WIMP events in particular. For example, the LF zenith angle should be higher than 68.7°. More information can be found in Ref. [1].

2.1.2 LowUp

The LowUp filter is again mainly designed for WIMP searches, but also atmospheric neutrino analysis and is mainly designed to capture up-going muons with an energy below 1 TeV. The majority of the events that are selected by this filter make use of the in-ice Volume Trigger (see Table **??**), but also the in-ice SMT8, in-ice String and SMT3-DeepCore triggers are run over for completeness. The selection cuts are loose selections required to look for up-going track-like particles. For example, the zenith angle of the reconstructed particle should have an angle of 80°

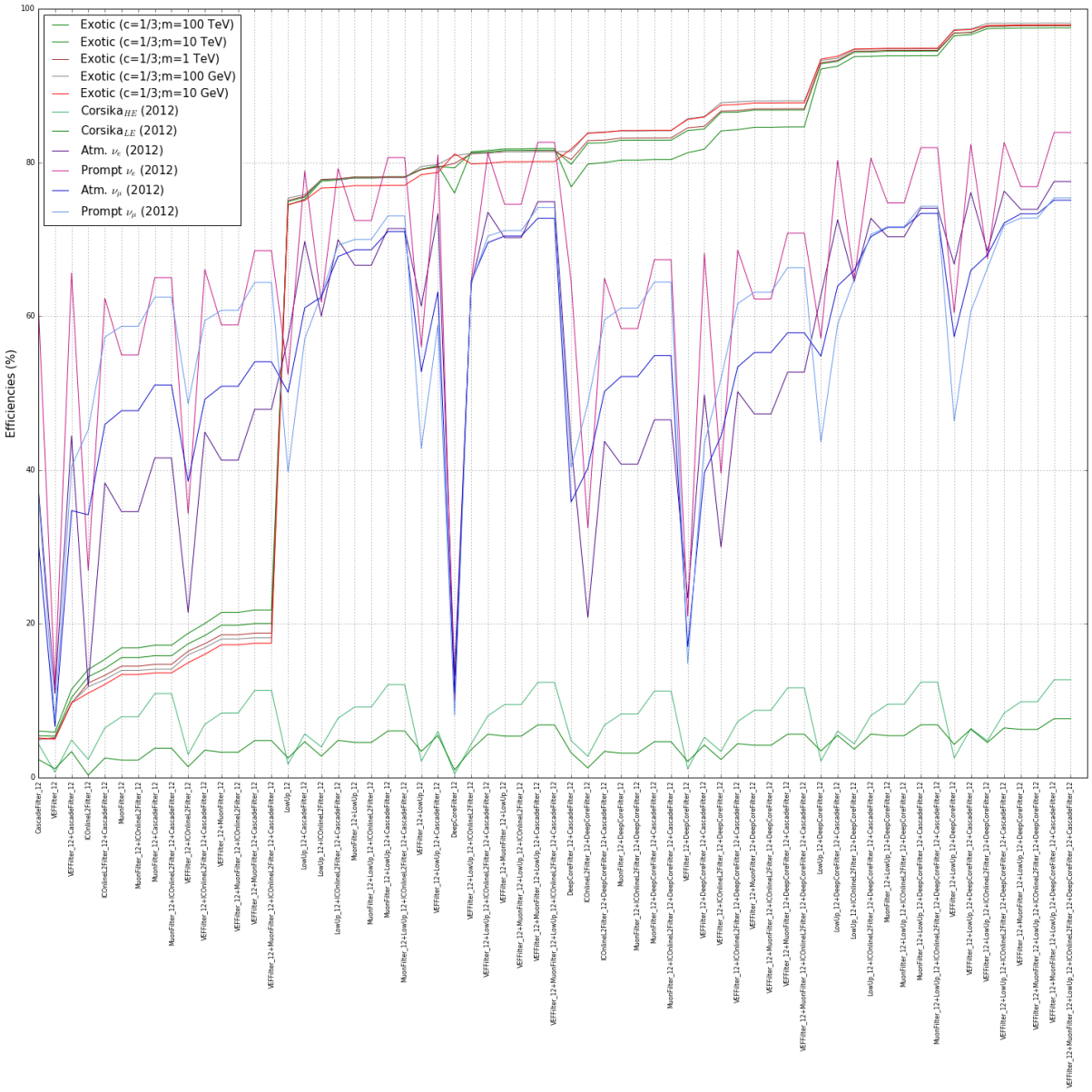


Figure 2.1: Illustration of the efficiencies of several filters and their possible combinations. The x-axis was determined by starting with filter selections that had a low efficiency in signal selection and range in function of performance. Five signal points for a fixed charge and different mass show similar results. Exotic SMPs with charges 1/2 and 2/3 show very similar results but are left out for a better visualization.

or higher and the difference between the maximal z-coordinate and minimal z-coordinate of hit DOMs should be less than or equal to 600 m. More information can be found in Ref. [2].

2.1.3 Online Muon L2

The Online Muon L2 filter is a subset of the Muon Filter (see Ref. **??**) and tries to select the most interesting muon-like events while reducing the rate of the filter from around 30 Hz to 5 Hz, reducing the data with a factor of 6. Historically this subset was processed data from the Muon Filter, but after realizing that this could be done online and because many analyses made use of this selection, it was chosen to implement it as a separate filter. The filter tries to select both up-going and down-going muons, with different selection cuts depending on the zenith angle of the particle reconstruction. The four selection ranges are defined as:

- $180° \geq \theta_{\text{MPE}} \geq 115°$
- $115° > \theta_{\text{MPE}} \geq 82°$
- $82° > \theta_{\text{MPE}} \geq 66°$
- $66° > \theta_{\text{MPE}} \geq 0°$

where the particle reconstruction was done with MPE (Section 1.1.3), which was feasible if it only had to be done on the events passing the Muon Filter. The first two regions have an efficiency* higher than 99%. The down-going region require more stringent cuts to remove the less interesting muons from air showers. The variables used are the number ot hit DOMs, likelihood parameters, number of PEs and so on. More information can be found in Ref. [3].

Verhoogt uw signaal niet zo veel omdat je enkel upgoing signaal gebruikte om dit te testen.

2.1.4 DeepCore

Additionally a DeepCore specialized filter was added to account for SMP tracks that partially traverse the more densly instrumented DC detector. Due to the low amount of light produced by these dim tracks, adding the DeepCore filter that is specialized for this part of the detector proved to be of significant importance.

The DeepCore filter was designed to look for very dim events coming from, e.g., dark matter, low-energy neutrino oscillations, and studies in observing atmospheric neutrinos below 100 GeV. The fiducial volume used for this filter consists of

- the bottom 22 DOMs on the IceCube strings 25, 26, 27, 34, 35, 36, 37, 44, 45, 46, 47 and 54;
- the bottom 50 DOMs on the DeepCore strings 79-86.

These strings are indicated in Fig. 2.2.

The filter uses the DeepCore SMT3 trigger and calculates the COG position. Two layers are used as a veto to remove events that probably originate from atmospheric muons. More information can be found in Ref. [4].

2.1.5 Burnsample checks

Before further processing, the burn sample (Section **??**) is compared over the different years that are used in the analysis. This is shown in Figure 2.3. More information on the burn sample can be found in Section **??**.

2.2 Level 3

The combined filter selection leads to a total rate of ~ 60 Hz, or ~ 1.9 billion events per year. The average event size at Level2 is around 15 kB, which would result into around 30 TB of data per year.

Therefore, five quality cuts are implemented with a goal that is threefold:

*Here defined as having a reconstruction within 3° of the MC truth.

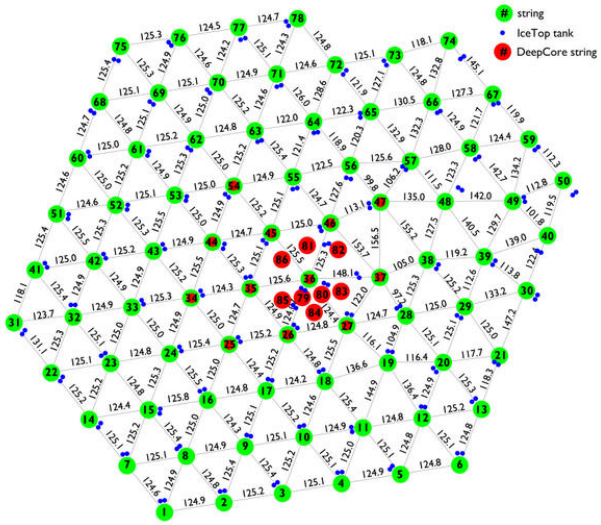


Figure 2.2: Aerial view of the IceCube strings (and IceTop tanks) where the DeepCore fiducial volume is defined by the DeepCore strings (red) and several surrounding in-ice IceCube strings (green and red).

1. reduce the total rate of the data,
2. improve the signal to background ratio, getting rid of uninteresting events,
3. improve the agreement between data and Monte Carlo.

These cuts are shown in Figs. 2.4, 2.5 and 2.6.

2.2.1 Zenith angle cut

Even though there are no up-going muons from air showers expected, the vast majority of events that pass the filter selection remain from misreconstructed muons. Even though there is only a small chance of these events to have a large misreconstructed zenith angle. The expected flux of air showers is so much larger compared to the assumed signal flux to such an extent that is dominates with orders of magnitude. The majority still has an reconstructed zenith angle lower than 90°. Therefore the zenith angle cut was set at an angle of

$$\theta_{\text{zen}}(\text{MPE}) \geq 85°. \tag{2.1}$$

2.2.2 RlogL cut

The reduced log-likelihood, rlogL of the track reconstruction fit is used as a goodness-of-fit variable. The term "reduced" is used because the logarithm of the likelihood is normalized by the number of degrees of freedom (NDOF) in the track fit

$$\text{rlogL} = \frac{\log \mathcal{L}}{\text{NDOF}} = \frac{\log \mathcal{L}}{\text{NCh} - \text{NPara}}, \tag{2.2}$$

where NCh is the number of channels/DOMs and NPara the number of fitted parameters (3 for the position and 2 for the track). For Gaussian probability distributions this expression corresponds to the reduced chi-square. Lower values indicate better reconstructions, therefore the rlogL cut was set at a value of

$$\text{rlogL} < 15. \tag{2.3}$$

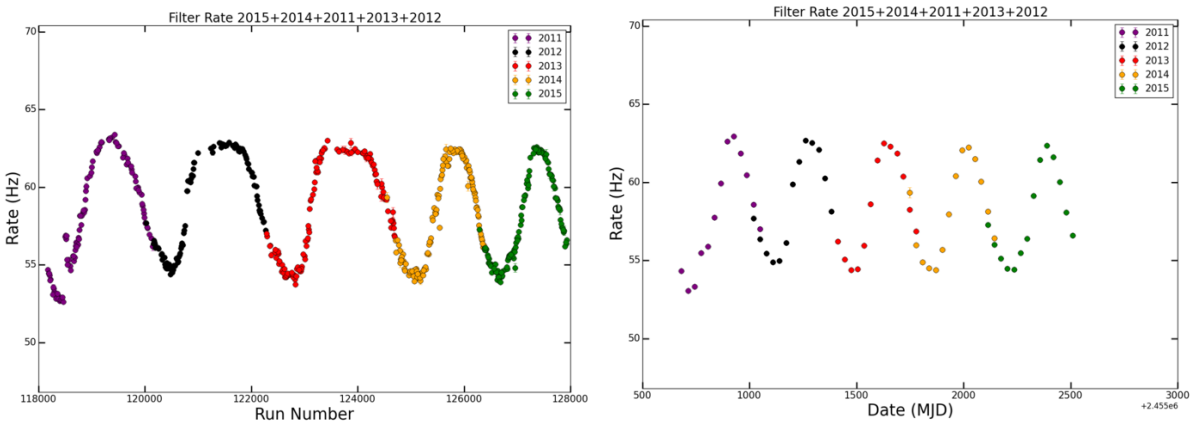


Figure 2.3: *Left:* Total rate of the combined filters in function of the run number. The sine wave pattern from seasonal variations in the atmosphere (see Section **??**) is clearly visible and consisten throughout the years. The x-axis is more spread out in the first years as there were more test runs. The shift in data rate in early 2011 runs is due to the DOM software change that was introduced in the Summer of 2011 [5]. This phenomenon is well understood and since the changes are minimal it was chosen to keep these runs. *Right:* Total filter rate averaged per month. There is an overlap for each year because a new season doesn't necessarily start in the beginning of a month.

2.2.3 NPE cut

The number of photoelectrons seen in the detector has a clear correlation to the number of photons that were emitted from the track. From Eq. **??** it is clear that particles with a charge < 1 will produce less light. Therefore a cut on the total number of photoelectrons was set at a value of

$$\text{NPE} < 50. \tag{2.4}$$

2.2.4 Starting rlogL cut

The relative probability for tracks to be starting and/or stopping can be computed with FiniteReco (see Section 1.1.5). Because most low-energetic muons would be starting and/or stopping in the detector, these likelihoods prove to be a powerful tool in removing these events *. The llh is always compared to the llh of throughgoing tracks, hence the "relative probability". It was chosen to place the starting rlogL at a value of

$$\text{rlogL} = \text{rlogL(starting)} - \text{rlogL(throughgoing)} > 0. \tag{2.5}$$

2.2.5 Stopping rlogL cut

Analogous to the previous cut, it was chosen to place the stopping rlogL at a value of

$$\text{rlogL} = \text{rlogL(stopping)} - \text{rlogL(throughgoing)} > 10. \tag{2.6}$$

2.3 Level 4

As can be seen in Figs. 2.4, 2.5 and 2.6, most of the background still originates from air showers (referred to as CORSIKA). Due to the Level 3 quality cuts, the total rate was reduced from around ~ 60 Hz to ~ 2 Hz, low enough for more elaborate variables to be computed and more elaborate cleaning. In Level 4 I have implemented the IceHive splitting and cleaning tools (see Section 1.3) and rerun the particle reconstructions on these "new" events. Additional quality

*High energetic muons will have a higher change of being throughgoing, but would produce much more light than the dim tracks that are expected for the SMPs.

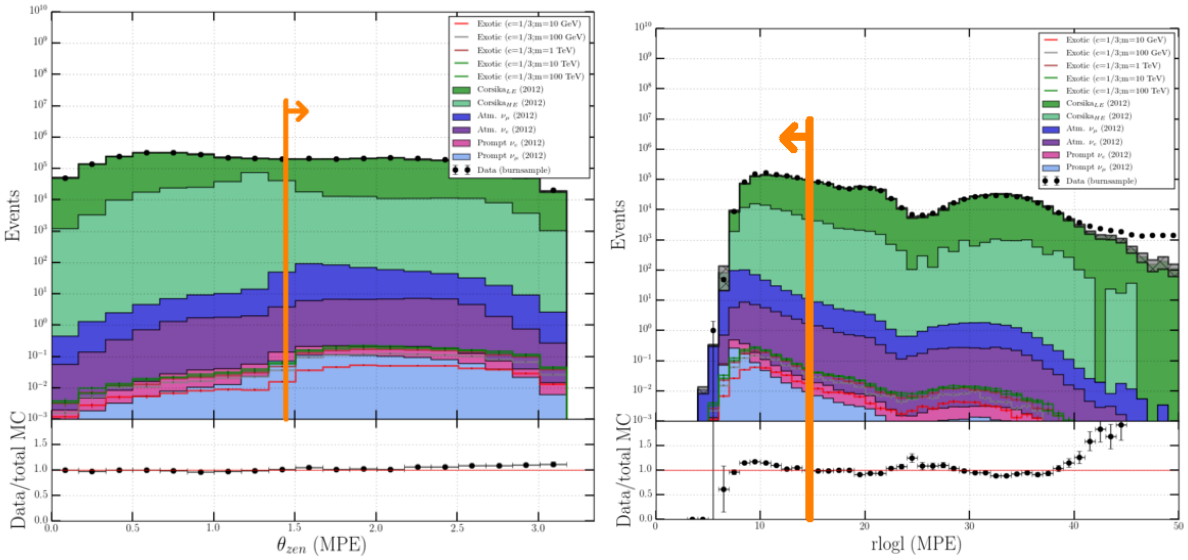


Figure 2.4: *Left:* Number of events in function of MPE reconstructed zenith angle normalized to the burn sample. The upwards trend to higher zenith angles is due to the filter selections that depend on the angle. *Right:* Number of events in function of rlogL normalized to the burn sample. The cuts are illustrated with an orange line, the arrow points towards the events that are kept.

Table 2.1: Overview of quality cuts in Level 4.

Variable	Definition	Cut	Motivation
nCh	Number of hit DOMs	≥ 5	Allows for better reconstructions
nStr	Number of hit strings	≥ 2	Allows for better reconstructions
nStr_in	The number of hit inner strings. An inner string is not located at the edge of the detector	≥ 1	Reduce leak-in events
Fitstatus MPE	Status of MPE reconstruction	Status == 'OK'	Remove bad reconstructions
θ_{HC} (MPE)	Zenith angle cut on HiveCleaned pulses	$\geq 85°$	Similar to cut explained in ???: focus on up-going tracks
Innerstring domination	See text inline	== True	See text inline

cuts were added to this level to ensure higher quality events. An overview is given in Table 2.1. Finally, new variables were constructed to use in Level 5.

2.3.1 Cleaning and quality cuts

IceHive provides for a thourough cleaning method, sometimes resulting into events with a very low amount of hit DOMs. However, a minimal amount of hits is required to have reasonable and trustworthy particle reconstructions. Similarly, more than one string should have a hit to allow for better reconstructions due to the sparse distribution of the strings in the detector. Because light is able to reach the edge of the detector, even if the closest approach of the particle is tens or hunderds of meters away for very bright events, is would be near impossible to distinguish bright events far from the detector to dim tracks passing close by. Therefore, it was required that at least one string not on the edge of the detector should have hit DOMs to reduce these *leak-in events*. The zenith angle cut is re-introduced on the new event that should have better reconstructions due to cleaning and finally there is a requirement for "innerstring domination".

Innerstring domination

There persist classes of events at the boundary of the detector, which can be a problem for an upgoing track analysis. This includes event classes as:

- (Leak in) Events which are heading towards the instrumented volume, but stop right before they reach it or pass next to, but not to far from, the detector. These leak light to the detector boundaries.
- (Boundary) Events that penetrate the detector very shallow on the boundary lines and

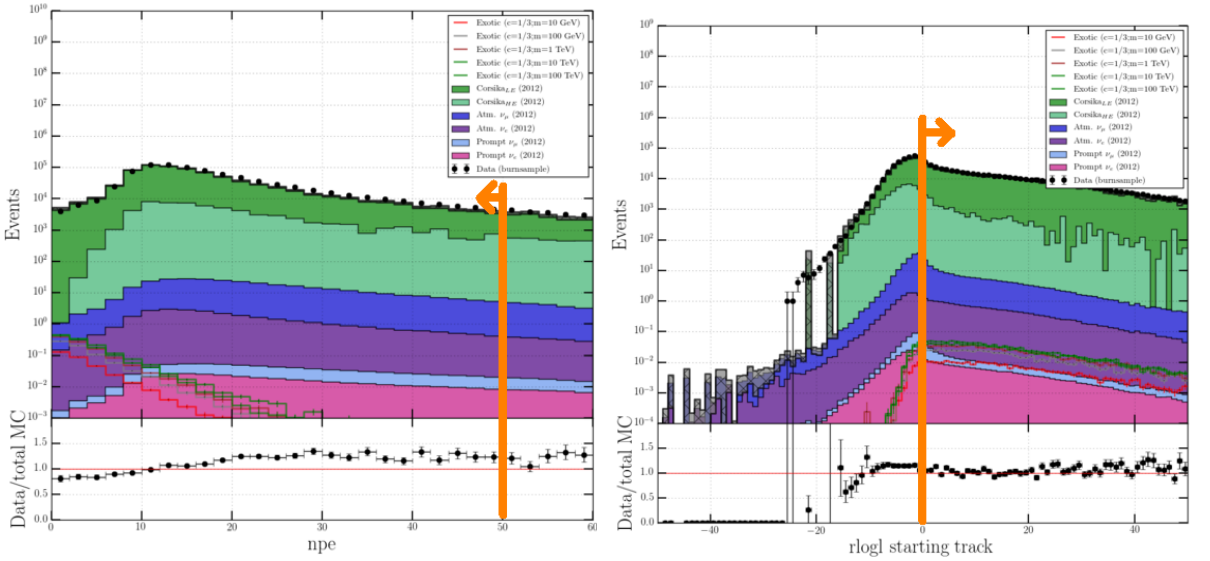


Figure 2.5: *Left:* Number of events in function of number of photoelectrons (NPE) seen in the detector. *Right:* Number of events in function of the starting likelihood. The cuts are illustrated with an orange line, the arrow points towards the events that are kept.

possibly have a cascade at the endpoint. The events have rather cascade like characteristics.
- (Corner-clippers) Events that are throughgoing on the corners of the detector that have a COG at a corner of the detector.
- (Leak out) Events originating from a neutrino passing through almost the entire length of the detector and only have an interaction vertex right before leaving the detector. Depending on position and angle, the reverse direction of reconstruction can be of similar probability and thus a nuisance.

All these event classes are not well reconstructable or have a high uncertainty in the reconstruction. It is more feasible to remove these class of events to maintain a sample of well reconstructable events. This is done here by the requirement of innerstring domination.

DOMs are defined as outer DOMs if they are one of the following:

- part of a string on edge of the detector,
- on the bottom of strings 1-78,
- on the top of strings 1-78.

The innerstring domination is set to `True` when

$$\frac{\#\text{outer DOMs}}{\#\text{inner DOMs}} < 0.5. \tag{2.7}$$

2.3.2 Variable construction

To distinguish signal from background events, variables that show a clear distribution difference prove to have the most discriminative power. In this part of the analysis, multiple new variables are introduced with this goal. Some variables used in Level 5 are already explained in the text and need no further introduction, they are shown in Fig. 2.17. A summary is given in Table **??**.

2.3.2.1 Commonvariables

Variables that were often used in analyses often had subtle differences between them, making them prone to be a cause of errors. Multiple variable were therefore combined into one project, called "Commonvariables". The variables used here can be subdivided into three categories: track characteristics, hit statistics and time characteristics and are summarized in Table 2.2. Their distributions are shown in Figs. 2.7, 2.8, 2.9 and 2.10.

Because DC and IC DOMs have different quantum efficiencies (see Section **??**), the pulses

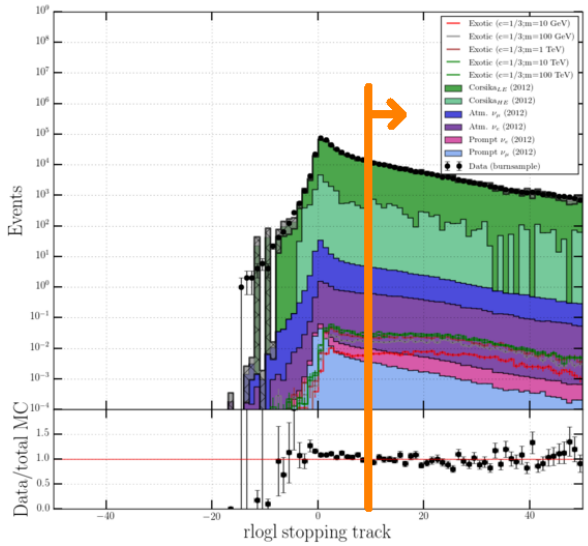


Figure 2.6: Number of events in function of the stopping likelihood. The cut is illustrated with an orange line, the arrow points towards the events that are kept.

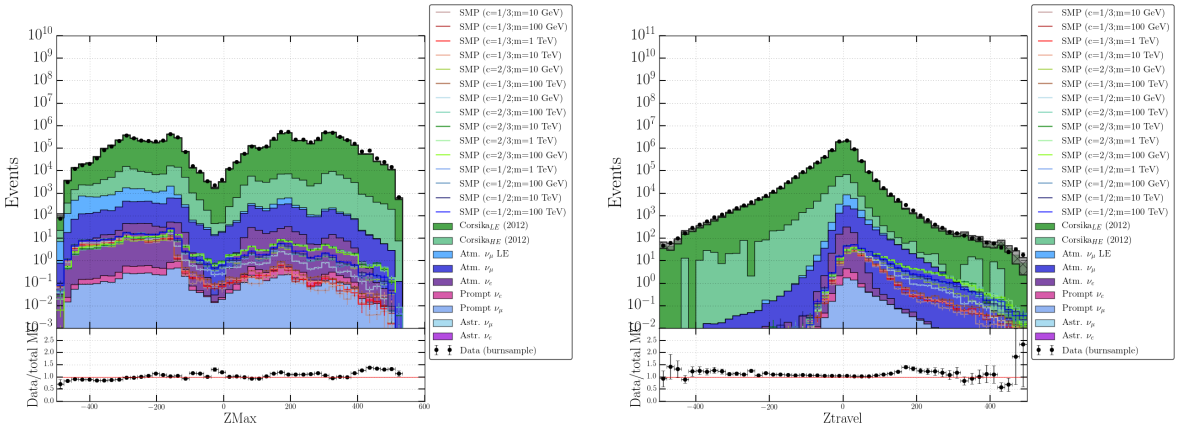


Figure 2.7: Blub

from DC and IC DOMs should not be mixed for an unambiguous definition. Therefore either only DC or IC pulses are used to compute these variables depending on if an event is *IC dominated* or *DC dominated*, where the former is set at $\frac{\#\text{DOMs}_{\text{IC}}}{\#\text{DOMs}_{\text{DC}}} >= 0.5$ and the latter otherwise.

2.3.2.2 Millipede variables

The `Millipede` toolkit was introduced in Section **??**, where it was explained how the energy deposition could be estimated from the light seen by the individual DOMs. Constructing multiple variables from this toolkit was the master thesis subject of Stef Verpoest and can be found in Ref. [6] for an elaborate explanation. The variables used in this analysis are explained below. Fig. 2.11 shows how the fit performed and can be helpful to explain the variables.

Mean loss

The most straightforward useage of the estimated mean energy loss rate is by taking the mean value and is referred to as *Mean_dEdX*. As can be seen from Fig. **??**, the distribution of SMPs peaks at lower values than known backgrounds as expected. Energy losses that are reconstructed to come from outside the detector are removed (hence the *_contained* in the figure).

Due to the squared charge dependencies, an SMP of charge 1/3 is expected to have a relative energy loss difference to muons with a factor of 9, which is the case when comparing to muons from neutrino interactions. Atmospheric muons in this level are almost entirely the result of

Table 2.2: List of Commonvariables used in this analysis.

† Whenever one of the track characteristics variables is shown/mentioned, the suffix (e.g. _50) refers to the track cylinder that was used around the track.

Category	Variable	Description
Track Characteristics†	AvgDistToDom	The average distance of the DOMs to the reconstructed track, weighted by the total charge of each DOM.
	EmptyHits	The maximal track length along the reconstructed track that got no hits within a cylinder around the track.
	TrackSeparation	Distance how far the COG positions of the first and the last quartile of the hits are separated from each other.
	TrackDistribution	The track hits distribution smoothness value [-1;1] shows how smooth the hits of the given pulse series within the specified track cylinder radius are distributed along the track.
Hit Statistics	ZTravel	Z value of first quartile (in time) of the hit DOMs is calculated. ZTravel is the average difference of the z value of all hit DOMs with the first quartile z value.
	ZMax	The maximum z of all hit DOMs.
Time Characteristics	ZPattern	All first pulses per DOM are ordered in time. If a DOM position of a pulse is higher than the previous ZPattern is increased with +1. If the second pulse is located lower in the detector ZPattern decreases with -1. In general this variable gives a tendency of the direction of a track.

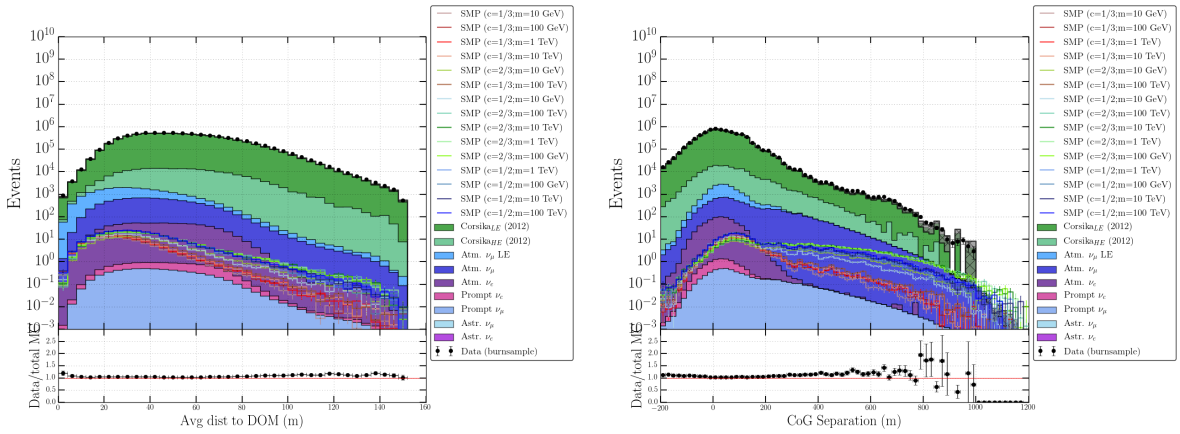


Figure 2.8: Blub

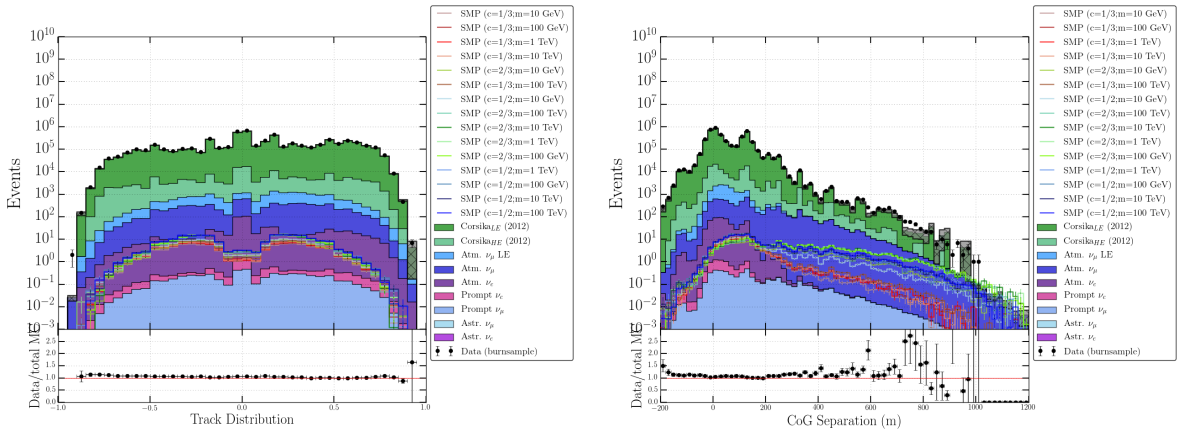


Figure 2.9: Blub

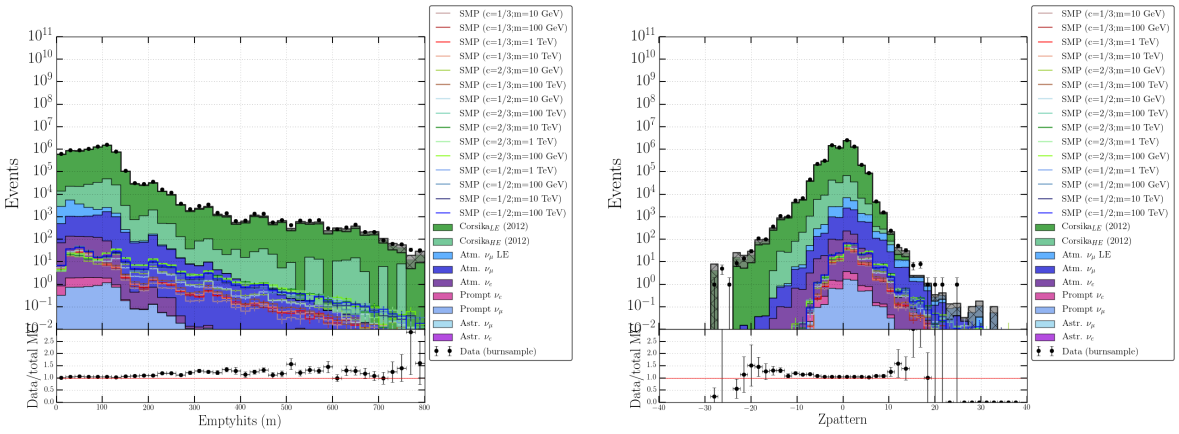


Figure 2.10: Blub

misreconstructions, corner clippers, etc. making a comparison not valid.

Uniformity

Once the mean is computed, it is possible to count the amount of times the energy distribution curve (red curve in Fig. 2.11) to cross the mean value. This variable therefore parametrizes the uniformity of the track. The reasoning behind distriminating signal events from background is that most SMPs will have less uniform triggered hits due to the low amount of light that is produced. Particles therefore need to travel closer to a DOM to trigger a hit. In Fig. 2.12 we can see that the background distributions peak at lower values than the signal, which also has a slower dropoff.

Track length

Because of the lower energy losses, the SMP particles are also expected to travel larger distances than muons, supporting the idea to contruct a variable that is sensitive to the distance traveled in the detector: a track length.

Since the tracks at this level are very dim, many segments from the `Millipede` output are reconstructed with zero energy. Therefore, it was chosen to use a certain part of the segments. The variables used here, *TrackLength_60*, used 60% of the length where energy was deposited. It is the distance between the points where 20% and 80% of the deposited energy track segment. We expected to see larger tails in the distribution of Fig. 2.13 in the signal compared to the background. Sub-optimal reconstructions result in bad millipede fits, giving a low and almost constant energy loss along the track. Additionally, coincident events that are not well separated also contribute to these events in the tail. However, most of the backgrounds events result into low values for this variable, still making it a powerful discriminative tool.

2.3.2.3 New variables

Speedratio

In addition, new variables were constructed. One was adopted from Jan Künnen's Earth WIMP analysis [7]. By looking at the "speed ratio" of the first to second and first to third HLC hits, it was used to remove wrongfuly simulated detector noise, helping in data/MC disagreement. In this analysis, it showed to provide for a modest addition to discriminating variables*. The *Speedratio* is defined as

$$\frac{v_{12}}{v_{13}} = \frac{d\left(\mathrm{HLC}_1, \mathrm{HLC}_2\right) / \Delta t\left(\mathrm{HLC}_1, \mathrm{HLC}_2\right)}{d\left(\mathrm{HLC}_1, \mathrm{HLC}_3\right) / \Delta t\left(\mathrm{HLC}_1, \mathrm{HLC}_3\right)}, \tag{2.8}$$

where $d\left(\mathrm{HLC}_i, \mathrm{HLC}_j\right)$ is the distance between the DOMs that recorded the ith and jth HLC hits. $\Delta t\left(\mathrm{HLC}_1, \mathrm{HLC}_2\right)$ is the difference in time of the ith and jth HLC hits. This distribution is

*Data and MC seem to agree well, which is probably due to the newer and better simulations.

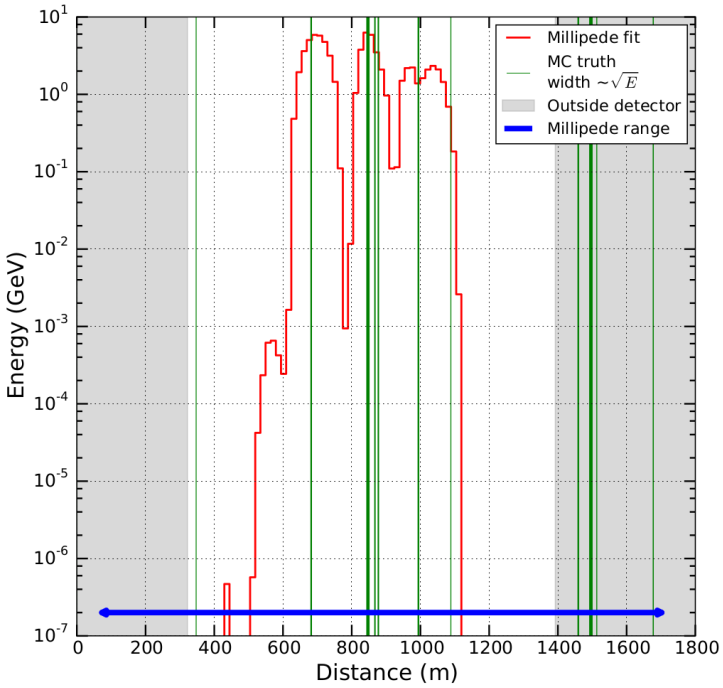


Figure 2.11: Output of a `Millipede` fit for an SMP with charge $\frac{1}{3}$ and mass 10 GeV. The x-axis shows the distance the particle traveled and starts after the first simulated energy loss event. The fit tries to estimate the energy of 15 m track segments. As a comparison, the true positions of energy deposits from the MC simulation are shown in green. Locations outside the detector are shaded in grey.

expected to peak at a value of 1, which is the expected result if one assumes that the photons originate from a particle traversing in a straight line and passes close to the DOMs. This is illustrated in Fig. 2.14.

NewLength

Because DC and IC pulses should not be mixed, essential information is lost regarding the length of a track. Many signal events are DC dominated (see Section 2.3.2.1), making those track length variables not optimal. This is especially the case for events that have hits in both IC and DC and are far away from each other, which is not expected from low-energetic muons. These should, on average, produce more light than SMPs and not travel very far unless they have significant energies that would result in much higher light outputs. I have constructed new variables that use the MPE track reconstruction as a seed. First, the event is required to have

- #DC pulses ≥ 4,
- #IC pulses ≥ 2,

since otherwise the contribution of noise infiltration is too high. Additionaly, pulses that lie within a cylinder with the seed track as the center are selected. The radius can be chosen, but is of the order of 50-150 meters. This radius is shown with a suffix after the variable (e.g. _100), if the radius is infinite (all pulses are used), the suffix is "_all".

Then the first/last quartile in DC hits and the first/last half of the IC hits are determined. From these one can easily calculate the COG. To determine a length from four COGs, two have to be selected. The selection is based on the timing information on these COGs and given in the Table 2.3.

All in all one can say that the NewLength variable is another attempt in defining the track length of the track in the detector. The suffix "_z" is used for the variable that only uses the z-coordinate. Negative values occur when the timing of the two selected COGs is inverted when compared to the seed track (e.g. if the seed track is downgoing but the first CoG in time

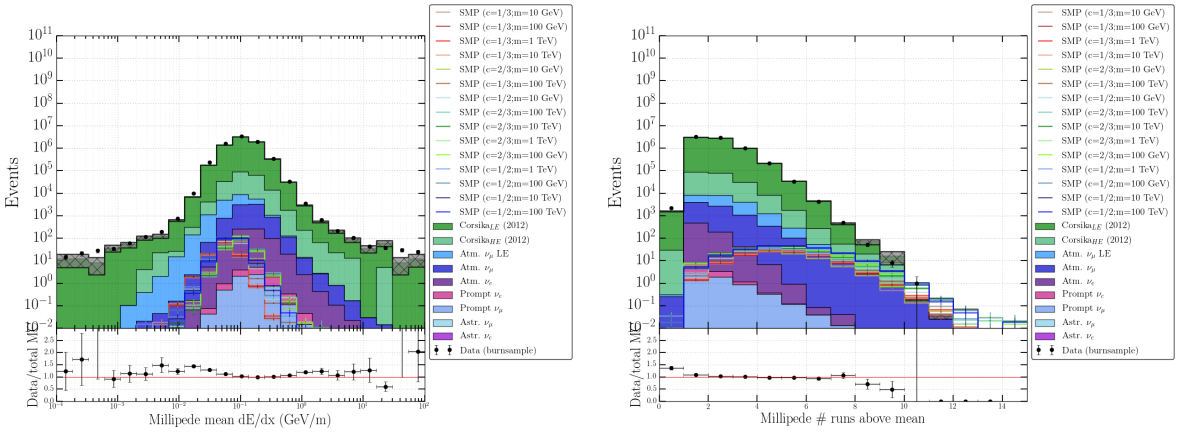


Figure 2.12: *Left:* Distributions for the estimated mean energy loss from the `Millipede` toolkit. *Right:* Distributions for the uniformity of the contained track.

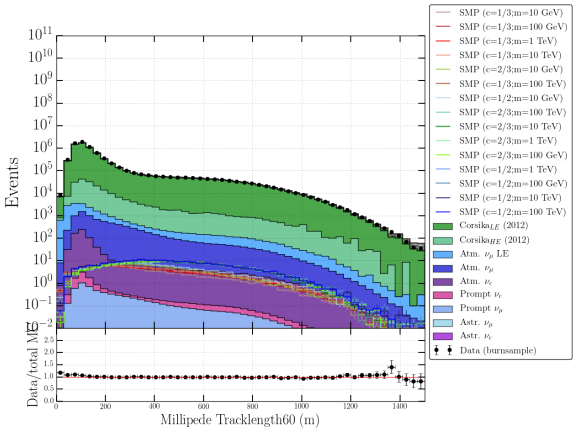


Figure 2.13: Distributions for the track length of the particle where the `Millipede` segments are used instead of the DOM pulses.

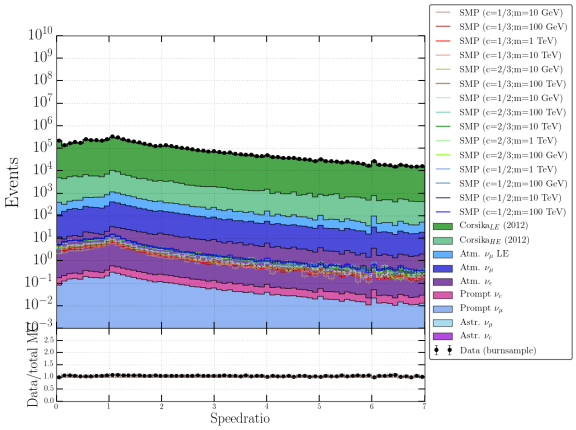


Figure 2.14: •

Table 2.3: •

Timing	COG_1
$\mathrm{DC}_{f,q} < \mathrm{IC}_{f,h}$	$\mathrm{DC}_{f,q}$
$\mathrm{DC}_{f,q} \geq \mathrm{IC}_{f,h}$	$\mathrm{IC}_{f,h}$
	COG_2
$\mathrm{DC}_{l,q} > \mathrm{IC}_{l,h}$	$\mathrm{DC}_{l,q}$
$\mathrm{DC}_{f,q} \leq \mathrm{IC}_{f,h}$	$\mathrm{IC}_{l,h}$

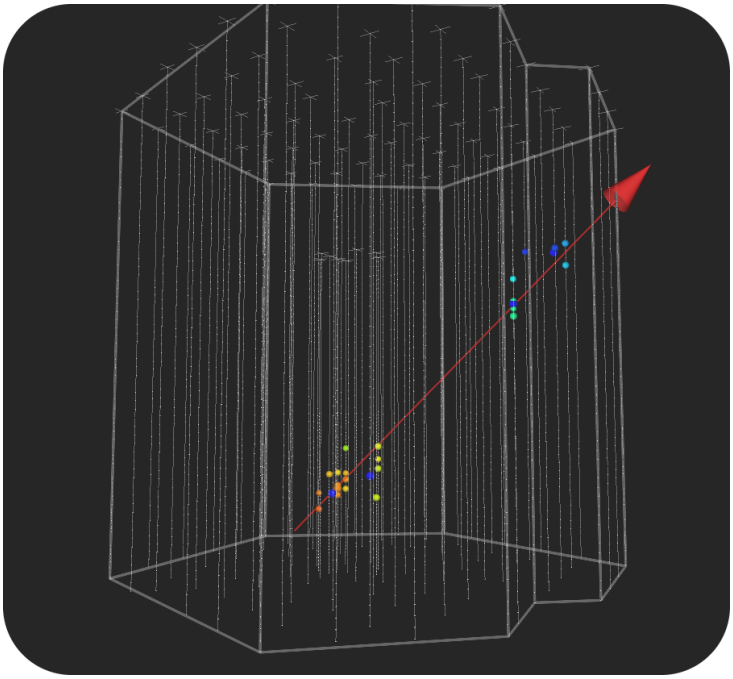


Figure 2.15: The NewLength variable is constructed by selecting the first quartile/half of the COGs of DC/IC and computing the distance from the last quartile/half of the COGs of DC/IC. Out of the four, the first and last in pulse time are chosen to compute the distance.

is located below the second COG) and most often occurs when the reconstruction was not optimal.

An illustration how the NewLength variable is constructed is shown in Fig. 2.15 and the distributions in Fig. 2.16.

2.3.3 Variable selection

The variables that are used in the BDT in Level 5 were selected by using the mRMR technique explained in Section 1.7. The 17 most important variables were used in the BDT. Less variables meant a lower performance while more variables did not show to add additional power in the BDT performance and meant more computational power. An overview of these variables is shown in Tab. 2.4.

2.4 Level 5

The last part of the analysis makes use of the variables that were constructed and the 17 that were estimated from the mRMR technique as the most powerful. First, the result from a single BDT is shown. Due to the lack of statistics in the final selection, the pullvalidation method was used for a limit computation.

2.4.1 BDT result

The parameters that were used for BDT training (see Section 1.6) are:

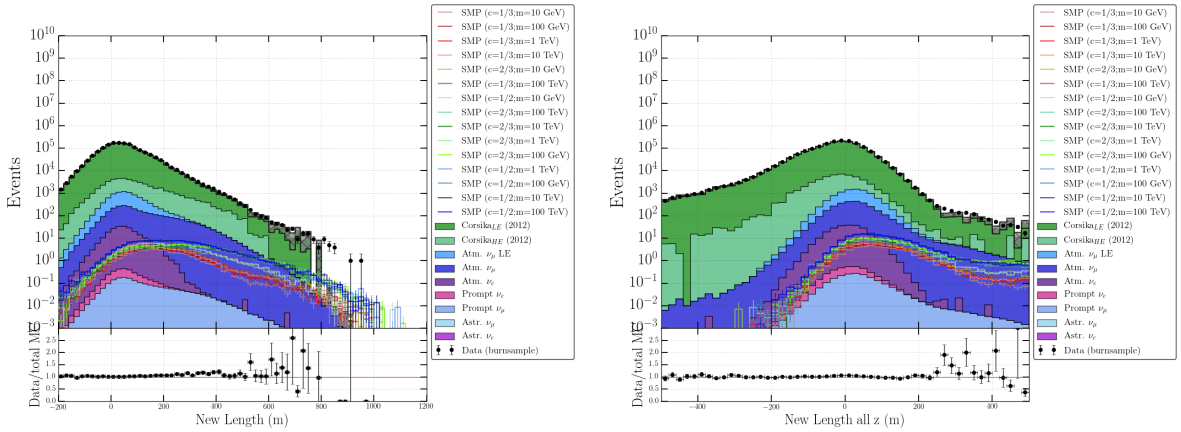


Figure 2.16: •

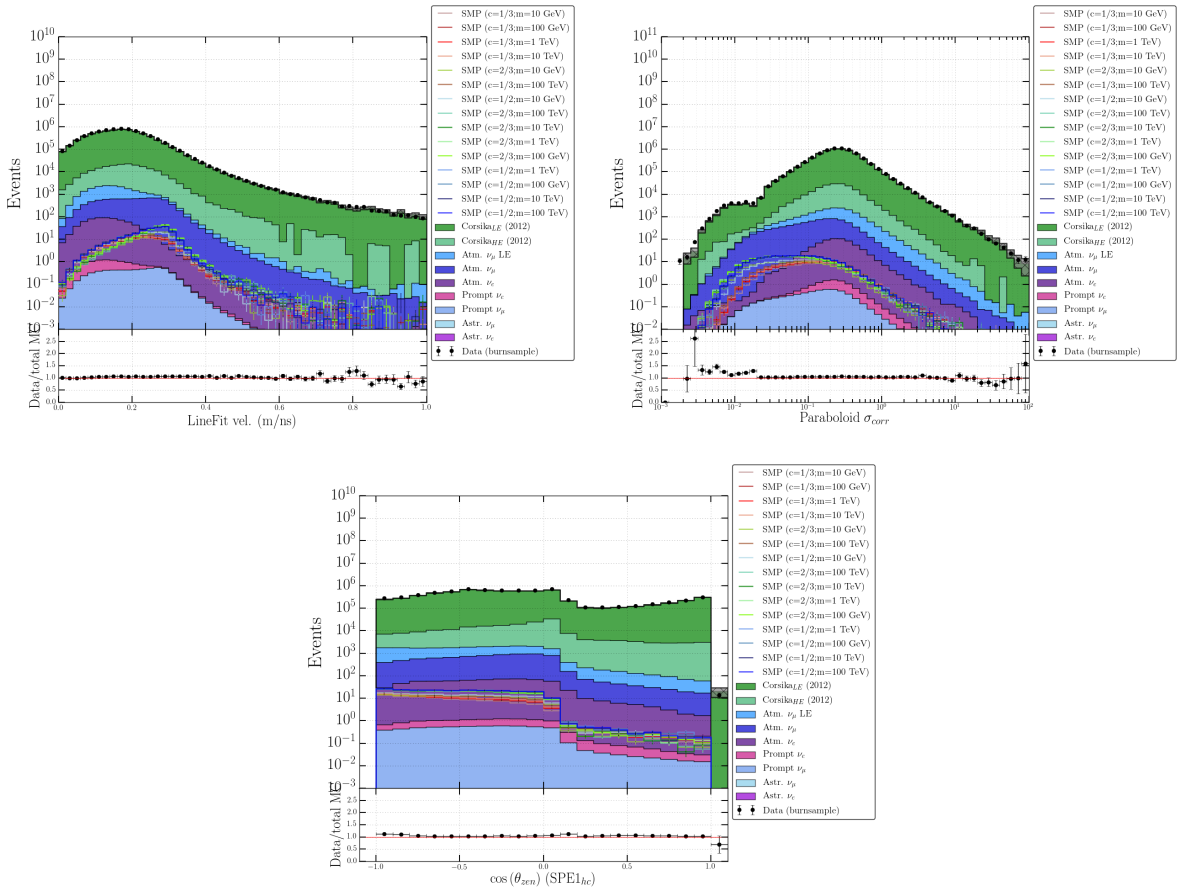


Figure 2.17: Blub

Table 2.4: My caption

Class	Variable	MRMR score	Importance
Commonvariables	ZMax	2	0.109
	ZTravel	3	0.106
	AvgDistToDom_150	9	0.048
	TrackSeparation_150	10	0.043
	TrackDistribution_50	12	0.035
	TrackSeparation_50	13	0.034
	EmptyHits_100	16	0.027
	ZPattern	17	0.016
Millipede	RunsAboveMean	4	0.105
	Mean_dEdX	5	0.074
	TrackLength_60	11	0.039
New variables	NewLength_150	1	0.132
	NewLength_all_z	6	0.059
	SpeedRatio	14	0.033
Other variables	LineFit_Velocity	7	0.055
	σ_{para}	8	0.051
	$\cos(\theta)_{\text{SPE}}$	15	0.033

MRMR score	Variable
1	NewLength_150
2	ZMax
3	ZTravel
4	RunsAboveMean
5	Mean_dEdX
6	NewLength_all_z
7	LineFit_Velocity
8	σ_{para}
9	AvgDistToDom_150
10	TrackSeparation_150
11	TrackLength_60
12	TrackDistribution_50
13	TrackSeparation_50
14	SpeedRatio
15	$\cos(\theta)_{\text{SPE}}$
16	EmptyHits_100
17	ZPattern

- Maximal depth: 2 (Fig. 1.8)
- Boosting β: 0.8 (Eq. 1.25)
- Number of trees: 400 (Section 1.6.3)
- Pruning factor: 35 (Section 1.6.4.1)

Training is done on 10% of the available burnsample as it showed to have a much better performance as opposed to training on background simulation due to the limited amount of CORSIKA simulation available. The other 90% of the burnsample is used for testing and is shown in the following plots. Testing on the MC datasets is also shown for the sake of completeness. Contribution of possible signal events, if they exist, in the data are minimal. Together with the very good data/MC agreement that is seen in the variables used, the training on data is a valid choice.

Also, it was chosen to select a (large) subsample of the signal to train the BDT to give the best possible results. One can see in the variable distributions in Figs. 2.10 and 2.16 that there are minimal contributions of events with negative ZTravel and/or negative NewLength values. Upgoing tracks should give positive values both and are therefore removed from the signal sample that is used to train the BDT*.

The result of one BDT can be seen in Fig. 2.18 and we can draw several conclusions:

- Data and MC show a very good agreement.
- The rate in background events is reduced with 4 to 5 orders of magnitude at the a BDT score around 0.25.
- At highere BDT scores, muons from low energetic muon neutrinos become a much more significant part of the total background than muons from air showers.
- The signal used to train the BDT is more concentrated at higher scores, as expected. The total signal sample and the subset used for training overlap at scores higher than 0.1.
- At the highest scores, where the signal dominates, it is clear that there is a lack of statistics in both CORSIKA simulations as the burn sample.

`https://arxiv.org/pdf/physics/0312102v1.pdf`
`https://arxiv.org/pdf/1310.1284.pdf`

Ook ergens een tabel maken met info over je data runs. Duidelijk maken wat de livetime is bv en ook zeggen van wanneer to wanneer een bepaalde run liep (2011: mei 2011- mei 2012)

Klaus zijn paper? `https://arxiv.org/abs/1806.05696`

*Of course, the final signal rate is computed from the full signal sample.

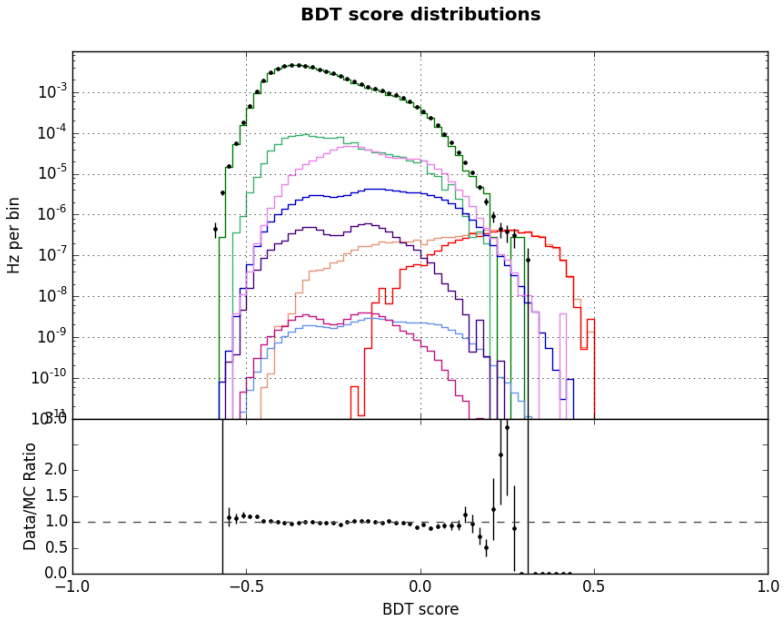


Figure 2.18: Blub

2.5 Pull validation

2.6 Systematic Uncertainties

2.7 Results

3. Summary and Discussion

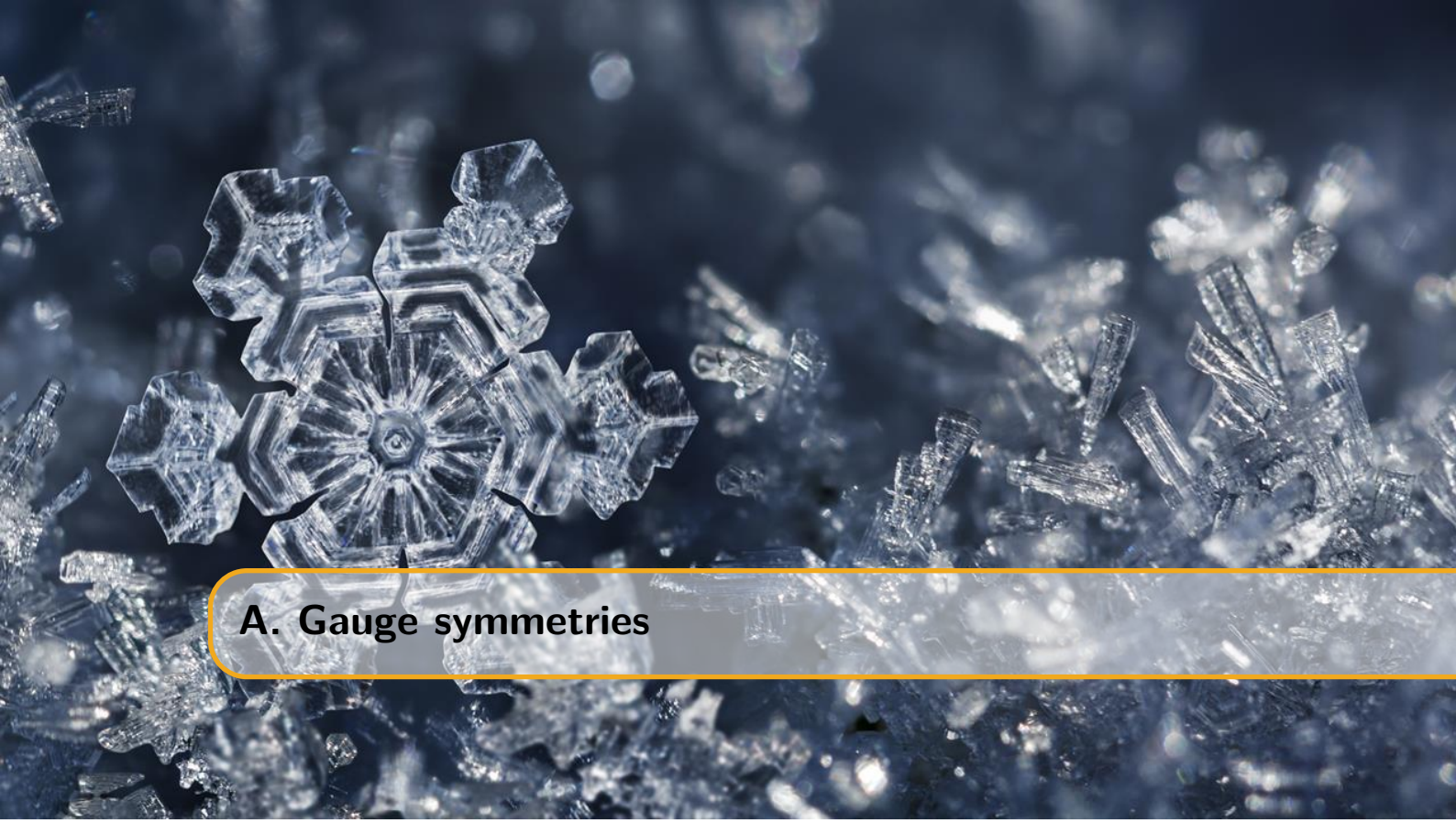
Possible improvements: filter! Trigger! a speed cut... If multiple COGs, connection of both should be within time window.

Other machine learning techniques.

Energy estimators (although probably wrong)

III Additions

Appendices



A. Gauge symmetries

NOG NIET GEDAAN

The difference between global and local symmetries are not straightforward for everybody. In this appendix I try to give a better view of the matter.

Imagine that at each point in space and time there is a circle attached to it. If one shifts all circles of all points with a fixed angle the underlying physics hasn't changed. If we look at the whole in a different angle, nothing seems to be changed as everything holds the same relative orientation. This is a global symmetry. For local symmeries we instead shift each circle through a different angle, but an angle that changes smoothly from point to point and in a way that we can say how that angle is varying between different nearby regions. Then it will turn out that we can describe that rotation angle by means of a so-called gauge field, which just lets us transport the charged scalar field from one point in space time to another, taking account of how the rotation angle of the circle is changing. A gauge is a kind of coordinate system that is varying depending on the location with respect to some underlying space. In physics we are almost always concerned with space-time as the underlying space, and we are typically interested in theories that are invariant with respect to the choice of gauge or coordinate system.

Dan wat uitleg vanuit je QFT boek en de dingen hieronder: Je wilt je derivative anders doen werken in je theory onder een transformatie, maar daarvoor heb je een veld nodig. M.a.w.: dankzij een veld heb je lokale ijktransformatie mogelijk!

B. Planck's law

bron: http://hyperphysics.phy-astr.gsu.edu/hbase/quantum/rayj.html

B.1 Electromagnetic waves in a cubical cavity

Suppose we have EM waves in a cavity at equilibrium with its surroundings. These waves must satisfy the wave equation in three dimensions:

$$\frac{\partial^2\Psi}{\partial x^2}+\frac{\partial^2\Psi}{\partial y^2}+\frac{\partial^2\Psi}{\partial z^2}=\frac{1}{c^2}\frac{\partial^2\Psi}{\partial t^2}. \tag{B.1}$$

The solution must give zero amplitude at the walls. A non-zero value would mean energy is dissipated through the walls which is in contradiction to our equilibrium assumption. A general solution takes the form of

$$\Psi(x,y,z,t)=\Psi_0\sin k_1x\sin k_2y\sin k_3z\sin k_4t, \tag{B.2}$$

which, after requiring $k_nL=n\pi$ with $n=0,1,2...$ and $k_4\frac{\lambda}{2c}=\pi$, leads to

$$\Psi(x,y,z,t)=\Psi_0\sin\left(\frac{n_1\pi x}{L}\right)\sin\left(\frac{n_2\pi y}{L}\right)\sin\left(\frac{n_3\pi z}{L}\right)\sin\left(\frac{2\pi ct}{\lambda}\right). \tag{B.3}$$

From the wave equation it is easy to find that

$$n^2=n_1^2+n_2^2+n_3^2=\frac{4L^2}{\lambda^2}, \tag{B.4}$$

which span up a sphere in "n-space" with a volume of $\frac{1}{8}\frac{4}{3}\pi n^{3/2}$, where the first term originates from the positive nature of $n_{1,2,3}$. Because there are two possible polarizations of the waves one has to multiply with an additional factor 2. The number of modes per unit wavelength is equal to

$$\frac{dN}{d\lambda}\times\frac{1}{L^3}=\frac{d}{d\lambda}\left[\frac{8\pi L^3}{3\lambda^3}\right]\times\frac{1}{L^3}=-\left[\frac{8\pi}{\lambda^4}\right]. \tag{B.5}$$

B.1.1 Classical approach

Following the principle of equipartition of energy, each standing wave mode will have an average enkergy kT with k the Boltzmann constant and T the temperature in Kelvin. The energy density is then:

$$\frac{du}{d\lambda} = -kT\frac{8\pi}{\lambda^4}. \tag{B.6}$$

In function of frequency $\nu = \frac{c}{\lambda}$:

$$\frac{du}{d\nu} = -\frac{c}{\lambda^2}\frac{du}{d\lambda} = \frac{8\pi kT\nu^2}{c^3}, \tag{B.7}$$

also known as the Rayleigh-Jeans law*. Problem: divergence

B.1.2 Quantum approach

The energy levels from a quantized harmonic oscillator are equal to

$$E_r = h\nu\left(r + \frac{1}{2}\right) = \frac{hc}{\lambda}\left(r + \frac{1}{2}\right) \quad \text{with} r = 0, 1, 2, ... \tag{B.8}$$

Implementing eq. B.4

$$E = \left(r + \frac{1}{2}\right)\frac{hc}{2L}\sqrt{n_1^2 + n_2^2 + n_3^2} \tag{B.9}$$

According to statistical physics the average energy is now not equal to kT but follows a probability distribution

$$p(\nu, r) = \frac{e^{\frac{-rh\nu}{kT}}}{\sum_{r=0}^{\infty} e^{\frac{-rh\nu}{kT}}}, \tag{B.10}$$

where we reference to the ground state of the oscillator: $E'_r = E_r - E_0$.

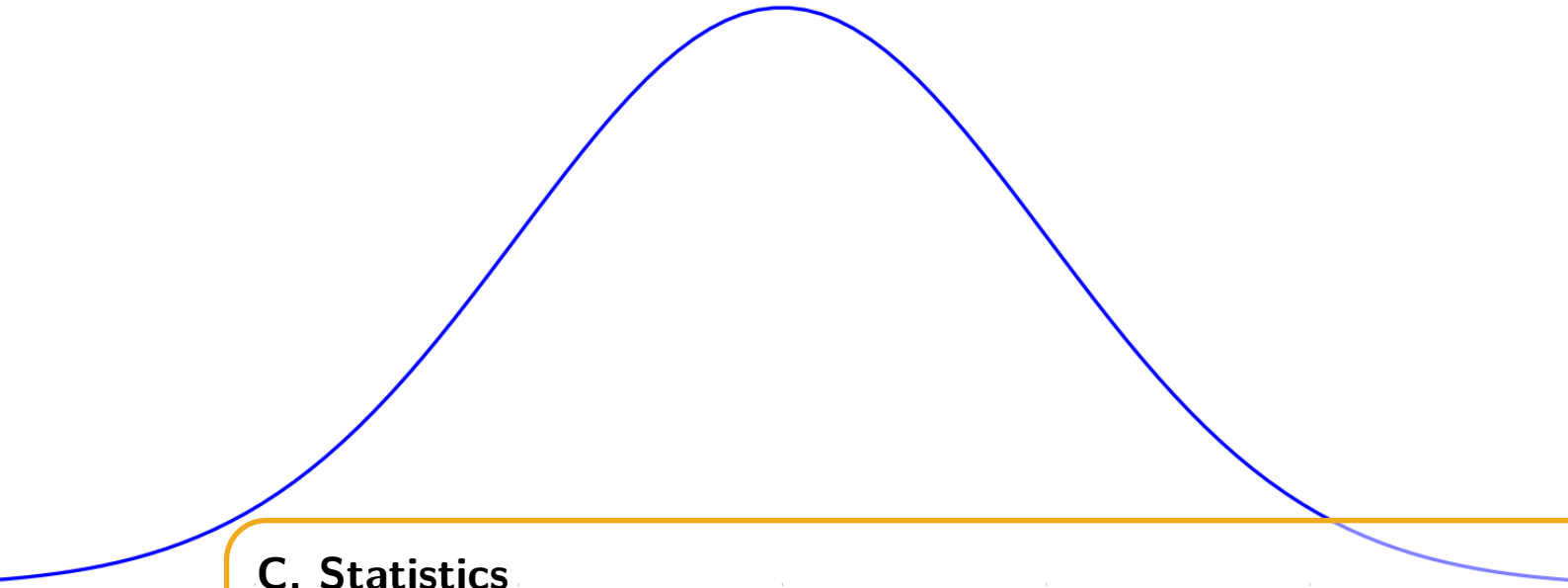
The average energy is now:

$$\begin{aligned}\langle E(\nu)\rangle &= \sum_{r=0}^{\infty} E(\nu, r)\cdot p(\nu, r) = \frac{\sum_{r=0}^{\infty} rh\nu\ e^{\frac{-rh\nu}{kT}}}{\sum_{r=0}^{\infty} e^{\frac{-rh\nu}{kT}}} \\ &= \frac{h\nu}{e^{h\nu/kT} - 1}\end{aligned} \tag{B.11}$$

Substituting this for kT in eq. B.7 we find Planck's equation:

$$\frac{du}{d\nu} = \frac{8\pi h\nu^3}{c^3}\frac{h\nu}{e^{h\nu/kT} - 1} \tag{B.12}$$

*This is often quoted per unit of staradian, which results in $\frac{2kT\nu^2}{c^3}$



C. Statistics

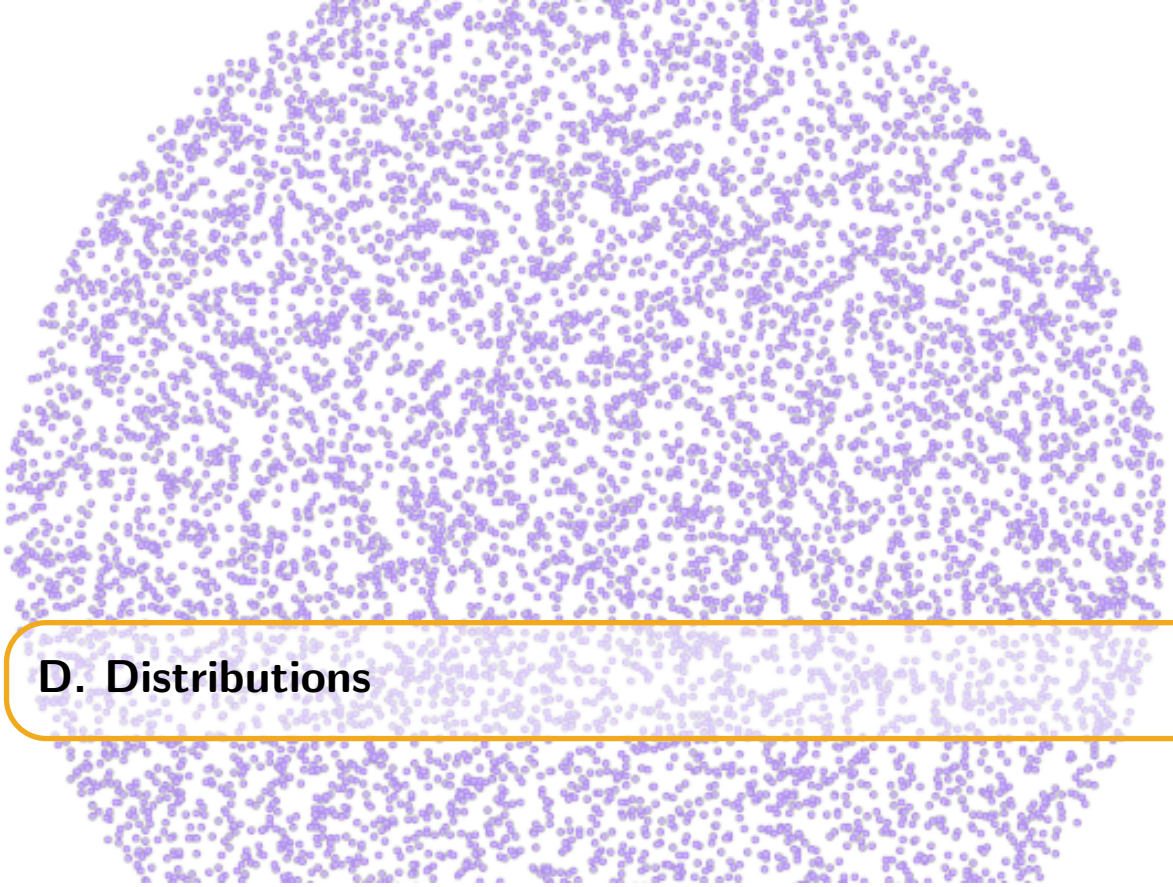
A word that is often mentioned in this work is “statistics”. It refers to the statistical error of a counting experiment, i.e. the Poissonian error. The Poisson distribution is a discrete probability of a certain number of $n$events occurring in a fixed time interval. The Poisson probability function is given by

$$P(n) = \frac{\lambda^n e^{-\lambda}}{n!}, \tag{C.1}$$

where λ is the expected number of events and also equal to the variance. An experiment that counted N events therefore has a statistical error of

$$\sigma = \sqrt{N} \tag{C.2}$$

In other words: higher statistics denotes a lower statistical error.



D. Distributions

D.1 Spherical random numbers

Most random number generators provide uniform distributions between the range $[0, 1]$. Assume we want to make a uniform distribution along a sphere with angles ϕ and θ and radius r, in spherical coordinates. Random numbers between $[0, \pi]$, $[0, 2\pi]$ and $[0, R]$ (the ranges of the coordinates) would not give a uniform distribution as illustrated in Fig. D.1 (left).
The differential surface area, dA, is equal to $dA(d\phi, d\theta) = r^2 \sin(\phi) d\phi d\theta$. If we want the distribution $f(v)$ to be constant for a uniform distribution, then $f(v) = \frac{1}{4\pi}$ since $\int \int_S f(v) dA = 1$ and $\int \int_S dA = 4\pi$. We want the distribution in function of the angles, so

$$f(v)dA = \frac{1}{4\pi} dA = f(r) f(\phi, \theta) d\phi d\theta. \tag{D.1}$$

Since we know the expression for dA, we find that

$$f(\phi, \theta) = \frac{1}{4\pi} \sin(\phi), \tag{D.2}$$

and separating the angles:

$$f(\theta) = \int_0^{\pi} f(\phi, \theta) d\phi = \frac{1}{2\pi}, \tag{D.3}$$

$$f(\phi) = \int_0^{2\pi} f(\phi, \theta) d\theta = \frac{\sin(\phi)}{2}, \tag{D.4}$$

where it is clear that $f(\phi)$ scales with $\sin(\phi)$; there are more points needed at the equator (this makes sense, as the surface at the equator is much larger!).

The question is now how one can get a sample to follow the distribution of $f(\phi)$. For this, we use the *inverse transform sampling* method where one makes use of the cumulative distribution function, $F(\phi)$, which increases monotonically

$$F(\phi) = \int_0^{\phi} f(\phi') d\phi' = \frac{1}{2} \left(1 - \cos(\phi)\right). \tag{D.5}$$

The method shows that if u is a random variable drawn from a uniform distribution, we have to find the inverse function of F,

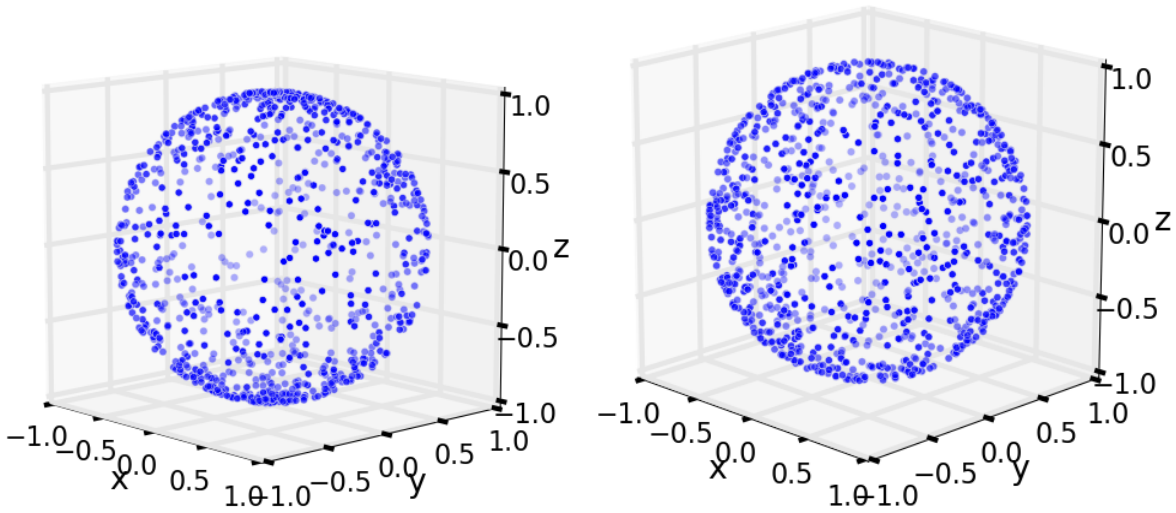


Figure D.1: *Left:* Illustration of a uniform sampling in angles ϕ and θ that doesn't give a uniform spherical distribution. *Right:* Illustration of a good spherical distribution.

$$F(F^{-1}(u)) = u \tag{D.6}$$

$$\frac{1}{2}\left(1 - \cos(F^{-1}(u))\right) = u \tag{D.7}$$

$$F^{-1}(u) = \arccos(1-2u). \tag{D.8}$$

In other words: if u is a random variable drawn from a uniform distribution, then $\phi = \arccos(1-2u)$ follows a distribution necessary for a uniform spherical distribution. Similarly, $\theta = \frac{1}{2\pi}u$.

D.2 Power law distributions

Analogous to what was written in the previous section, one can produce a power law distribution from random numbers using the inverse transform sampling method:

$$\begin{aligned} f(E) &= A \cdot E^{-\gamma} \quad \text{(powerlaw)} \\ F(E) &= \int_{E_{min}}^{E} A \cdot E^{-\gamma} dE = u \quad \text{(inverse sampling, u random number [0,1])} \\ &= A\left[\frac{E^{-\gamma+1}}{-\gamma+1}\right]_{E_{min}}^{E} \\ &= \frac{A}{-\gamma+1}\left(E^{-\gamma+1} - E_{min}^{-\gamma+1}\right) \end{aligned} \tag{D.9}$$

Because we know that $F\left(F^{-1}(u)\right) = u$, we can find an expression for $F^{-1}(u)$:

$$\begin{aligned} u &= \frac{A}{-\gamma+1}\left(\left(F^{-1}(u)\right)^{-\gamma+1} - E_{min}^{-\gamma+1}\right) \\ &\Rightarrow \\ F^{-1}(u) &= \left(\left(\frac{-\gamma+1}{A} \cdot u\right) + E_{min}^{-\gamma+1}\right)^{1/(-\gamma+1)} \end{aligned} \tag{D.10}$$

To find A, we use the property of a CDF:

$$F(E_{max}) = 1 \Rightarrow A = \frac{-\gamma+1}{E_{max}^{-\gamma+1} - E_{min}^{-\gamma+1}}, \tag{D.11}$$

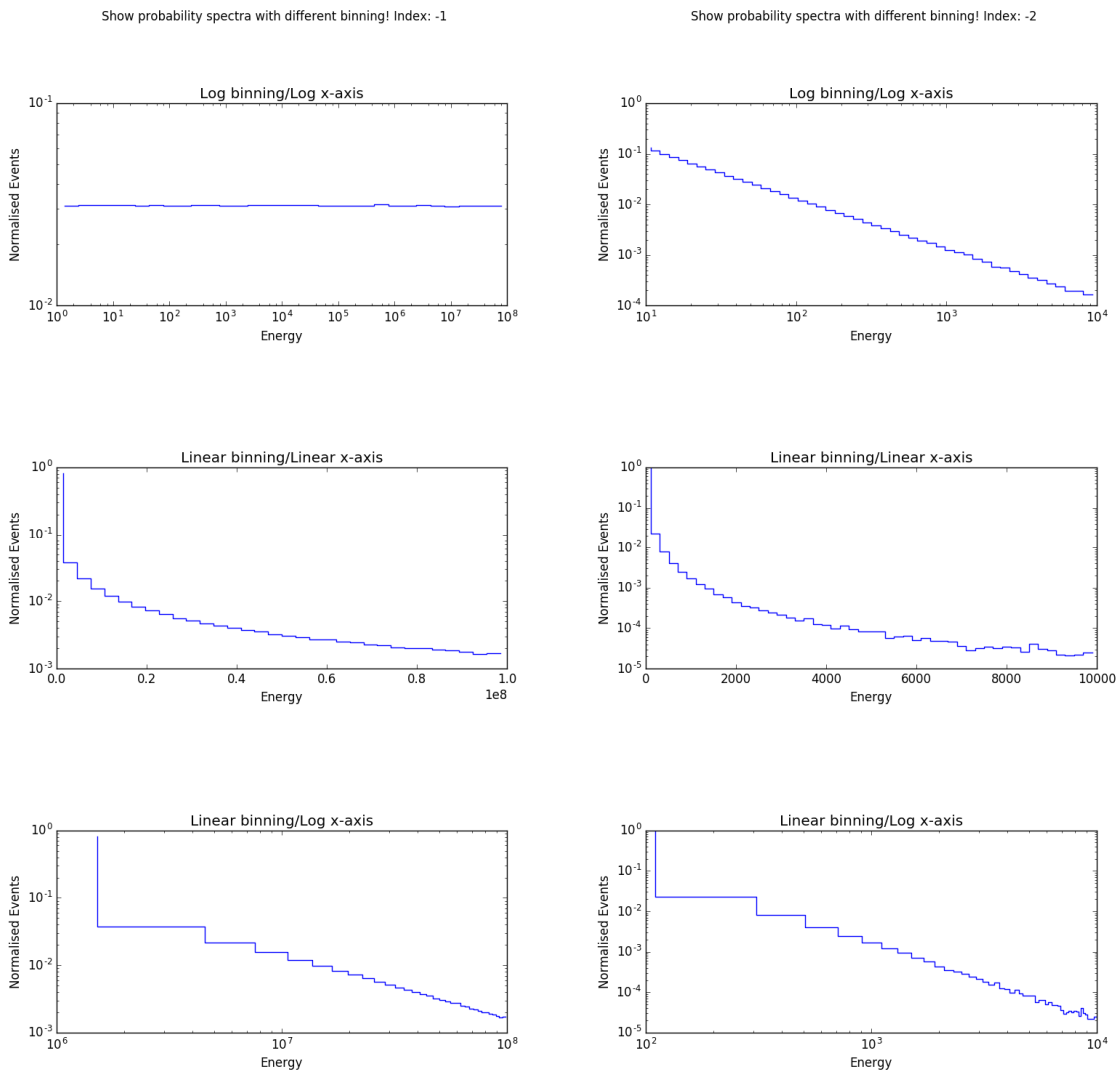


Figure D.2: *Left:* Histograms with different binnings showing the behavior of an energy spectrum with spectral index -1. *Right:* Histograms with different binnings showing the behavior of an energy spectrum with spectral index -2.

leading to

$$F^{-1}(u) = \left((1-u)\cdot E_{min}^{-\gamma+1} + u\cdot E_{max}^{-\gamma+1}\right)^{1/(-\gamma+1)}, \tag{D.12}$$

which shows how one can draw a distribution in function of E following $f(E)$ with a uniform random number u.

For γ = -1, the computations are analogous and one can see that this will produce a uniform distribution in log space. This is shown in Fig. D.2.

$$\begin{aligned} E &= E_{min}\cdot 10^{u\cdot\log\frac{E_{max}}{E_{min}}} \\ &= 10^{u[\log E_{min},\log E_{max}]} \end{aligned} \tag{D.13}$$

In Fig. D.3 the signal reweighting is shown.

D.3 Angular distributions

As seen in Section D.1, the differential space angle $d\Omega$ is equal to

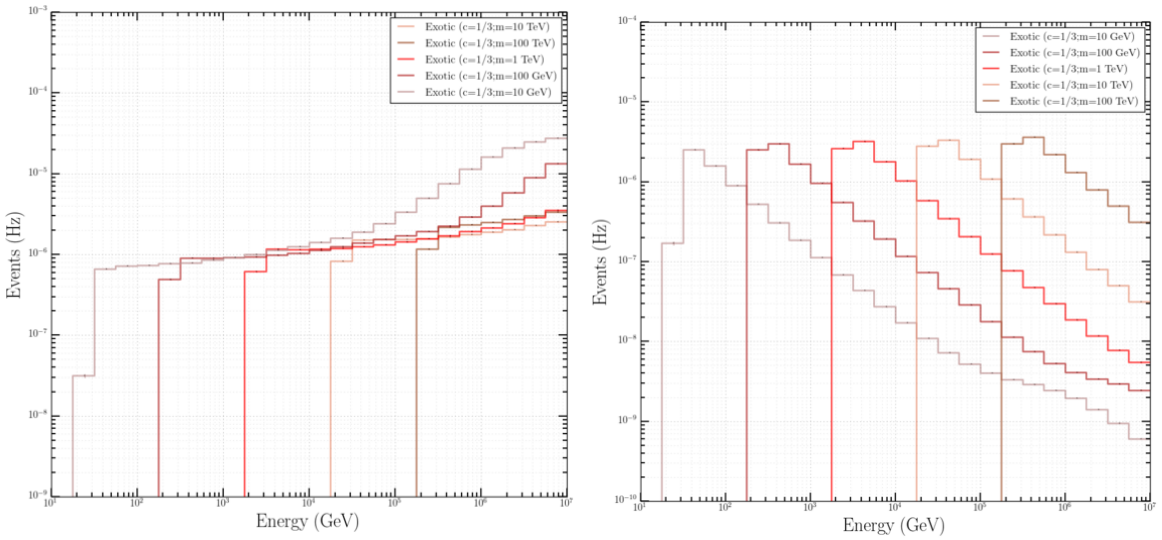


Figure D.3: *Left:* Spectrum of the signal before weighting following an E^{-1} spectrum. The rise in the rate in function of energy is due to the trigger efficiency that increases in function of energy. *Right:* Spectrum of the signal after reweighting to an energy spectrum of $E-2$.

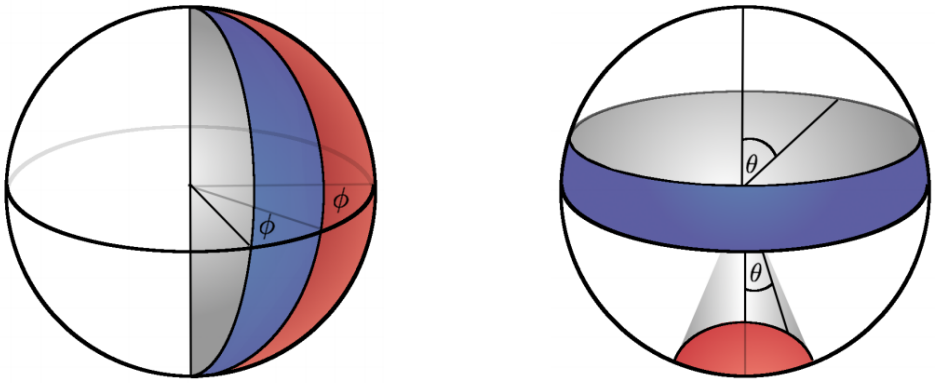


Figure D.4: Illustration of angle distributions in spherical coordinates. The blue and red surfaces are equal in size. The left figure clearly shows the surface to be proportional to the azimuth. The right figure shows how there is a non-trivial dependence on the zenith angle for equal partitions on the surface of a sphere.

$$d\Omega = \sin(\theta) d\theta d\phi. \tag{D.14}$$

If one shows the distribution of ϕ and/or θ, then this is the same as showing partial integrations per bin. We find that

$$\Omega \propto \cos(\theta), \tag{D.15}$$

or in other words: the space angle is proportional to the azimuth and the cosine of the zenith. An example is shown in Fig. D.4.

D.4 Weighting

A method that is often used in simulations is *weighting*. The simulated and expected differential flux of particles is often not the same, mainly due to two reasons:

- The flux has no uniform power law behavior. As can bee seen in Fig. **??**, there can be multiple "kinks" and changes in a spectrum. Instead of simulating the flux according to

one model, a general uniform flux is used and later reweighted to be able to fit to other models more easily.

- A steep power law indicates very few events at the highest energy bins. This means large CPU time would be necessary to simulate these events. As an example, let us assume two different fluxes

$$f_1 = A \cdot x^{-1}, \tag{D.16}$$

$$f_2 = B \cdot x^{-2}, \tag{D.17}$$

where $A = 10^3$ and $B = 10^4$, so the fluxes cross at a value of $x^{-1+2} = x = \frac{10^4}{10^3} = 10$. In the interval $x \in [10^3, 10^4]$, the number of events for f_1 is equal to 10^3, whereas for f_2 this is equal to 9.

Simulating with harder spectra* leads to more statistics in high-energy bins.

The weights can be generally written down as

$$w = \frac{dN_{exp}}{dAd\Omega dEdt} \times \frac{dAd\Omega dE}{dN_{sim}}. \tag{D.18}$$

A disadvantage of using weights is that certain events with a high weight are rare but can dominate or obscure the sample in the tails of certain distributions.

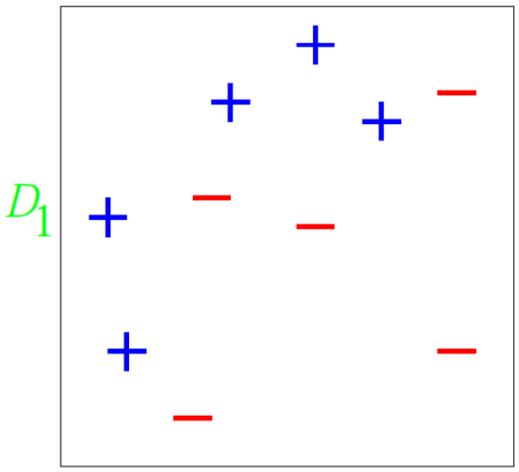
*Harder spectra equals to a lower gamma, since there will be more high-energy events.

E. AdaBoost: simple example

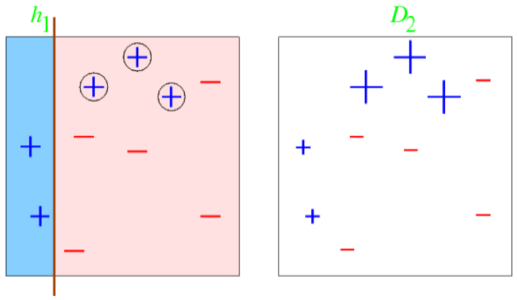
Consider a binary decision tree classification with 10 training examples. The illustrations below are 2D variable distributions.

We give each event an equal weight, making the weight distribution D_1 uniform. For this simple example, each of our classifiers will be an axis-parallel linear classifier (simple cut in one of the two variables).

Initial distribution



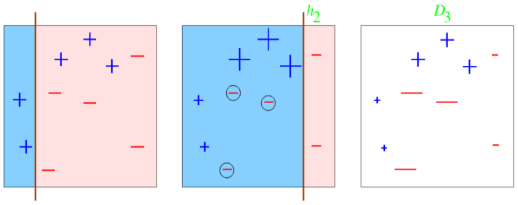
Round 1



- Error rate of h_1: $\epsilon_1 = 0.3$; weight of h_1 (see Eq. 1.25): $\alpha_1 = \frac{1}{2}\ln\left(\frac{1-\epsilon_1}{\epsilon_1}\right) = 0.42$

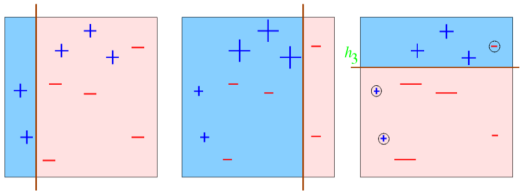
- An event that is misclassified gets a higher weight: weight multiplied with $\exp(\alpha_1)$
- An event that is correcly classified gets a lower weight: weight multiplied with $\exp(-\alpha_1)$

Round 2



- Error rate of h_1: $\epsilon_1 = 0.21$; weight of h_2 (see Eq. 1.25): $\alpha_2 = \frac{1}{2}\ln\left(\frac{1-\epsilon_2}{\epsilon_2}\right) = 0.65$
- An event that is misclassified gets a higher weight: weight multiplied with $\exp(\alpha_2)$
- An event that is correcly classified gets a lower weight: weight multiplied with $\exp(-\alpha_2)$

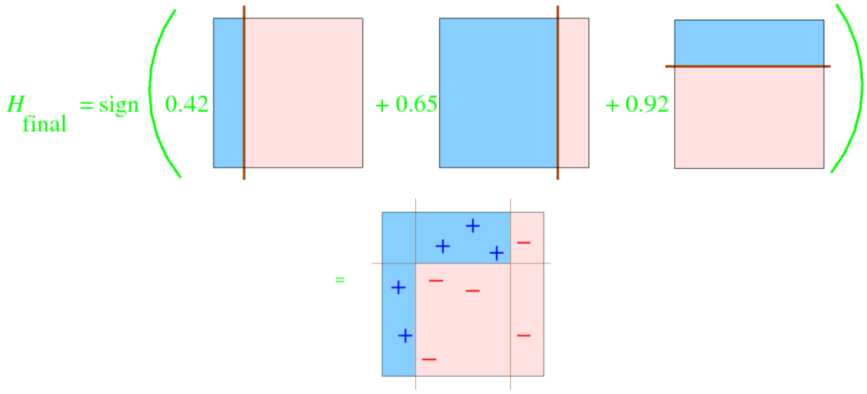
Round 3

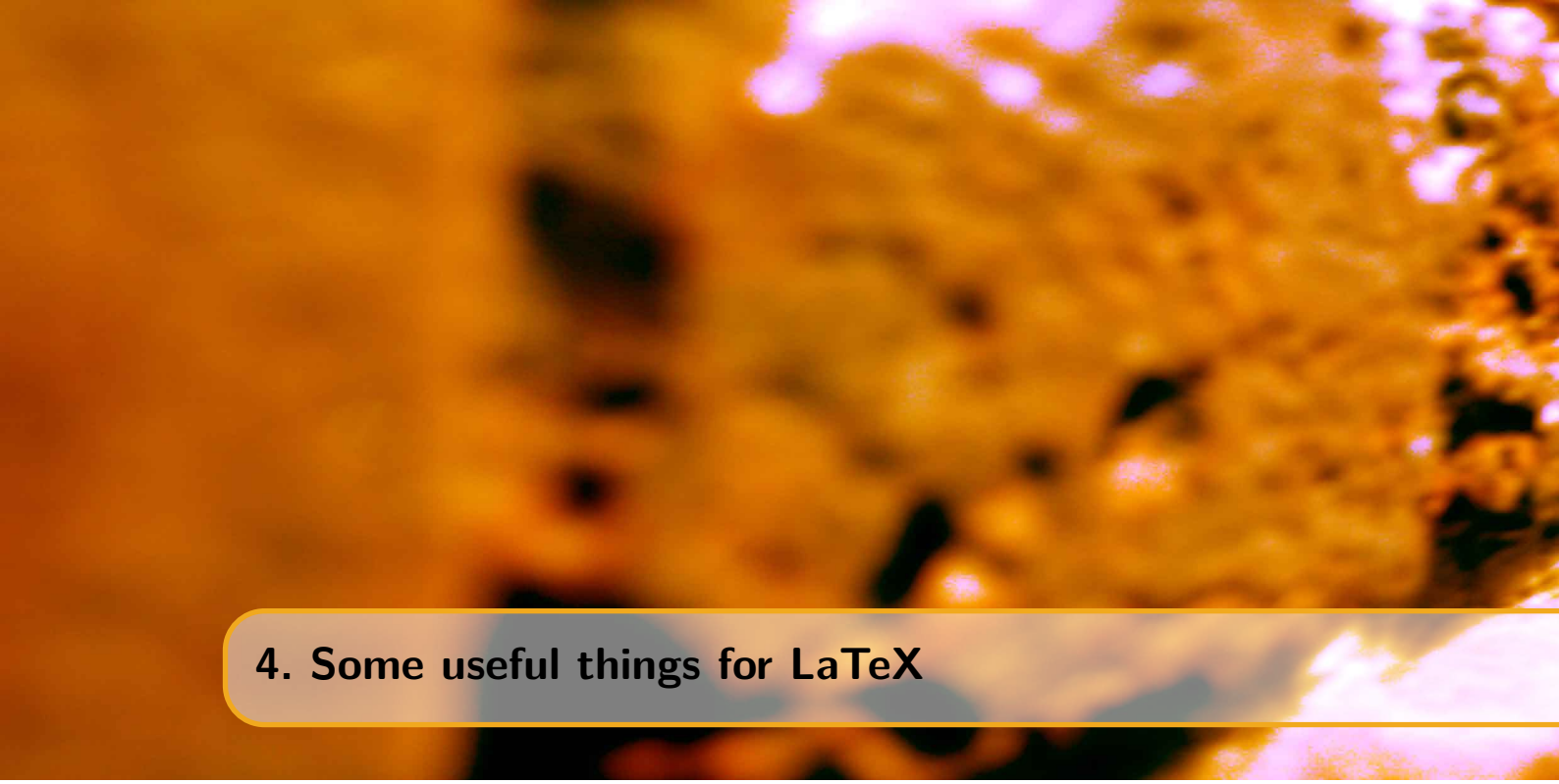


The error rate of h_1: $\epsilon_1 = 0.21$; weight of h_2 (see Eq. 1.25): $\alpha_2 = \frac{1}{2}\ln\left(\frac{1-\epsilon_2}{\epsilon_2}\right) = 0.65$
Let us suppose to stop after this round, we now have a forest of 3 decision classifiers: h_1, h_2, h_3.

Final step

The final classifier is a weighted linear combination of all the classifiers:





4. Some useful things for LaTeX

4.1 Definitions

This is an example of a definition. A definition could be mathematical or it could define a concept.

Definition 4.1.1 — Definition name. Given a vector space E, a norm on E is an application, denoted $||\cdot||$, E in $\mathbb{R}^+ = [0, +\infty[$ such that:

$$||\mathbf{x}|| = 0 \;\Rightarrow\; \mathbf{x} = \mathbf{0} \tag{4.1}$$

$$||\lambda\mathbf{x}|| = |\lambda| \cdot ||\mathbf{x}|| \tag{4.2}$$

$$||\mathbf{x} + \mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}|| \tag{4.3}$$

4.2 Remarks

This is an example of a remark.

R The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

4.3 Corollaries

This is an example of a corollary.

Corollary 4.3.1 — Corollary name. The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

4.4 Propositions

This is an example of propositions.

4.4.1 Several equations

Proposition 4.4.1 — Proposition name. It has the properties:

$$|||\mathbf{x}|| - ||\mathbf{y}||| \leq ||\mathbf{x} - \mathbf{y}|| \tag{4.4}$$

$$||\sum_{i=1}^{n} \mathbf{x}_i|| \leq \sum_{i=1}^{n} ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \tag{4.5}$$

4.4.2 Single Line

Proposition 4.4.2 Let $f, g \in L^2(G)$; if $\forall \varphi \in \mathcal{D}(G)$, $(f, \varphi)_0 = (g, \varphi)_0$ then $f = g$.

4.5 Examples

This is an example of examples.

4.5.1 Equation and Text

■ **Example 4.1** Let $G = \{x \in \mathbb{R}^2 : |x| < 3\}$ and denoted by: $x^0 = (1, 1)$; consider the function:

$$f(x) = \begin{cases} \mathrm{e}^{|x|} & \text{si } |x - x^0| \leq 1/2 \\ 0 & \text{si } |x - x^0| > 1/2 \end{cases} \tag{4.6}$$

The function f has bounded support, we can take $A = \{x \in \mathbb{R}^2 : |x - x^0| \leq 1/2 + \epsilon\}$ for all $\epsilon \in]0\,; 5/2 - \sqrt{2}[$. ■

4.5.2 Paragraph of Text

■ **Example 4.2 — Example name.** Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

■

4.6 Exercises

This is an example of an exercise.

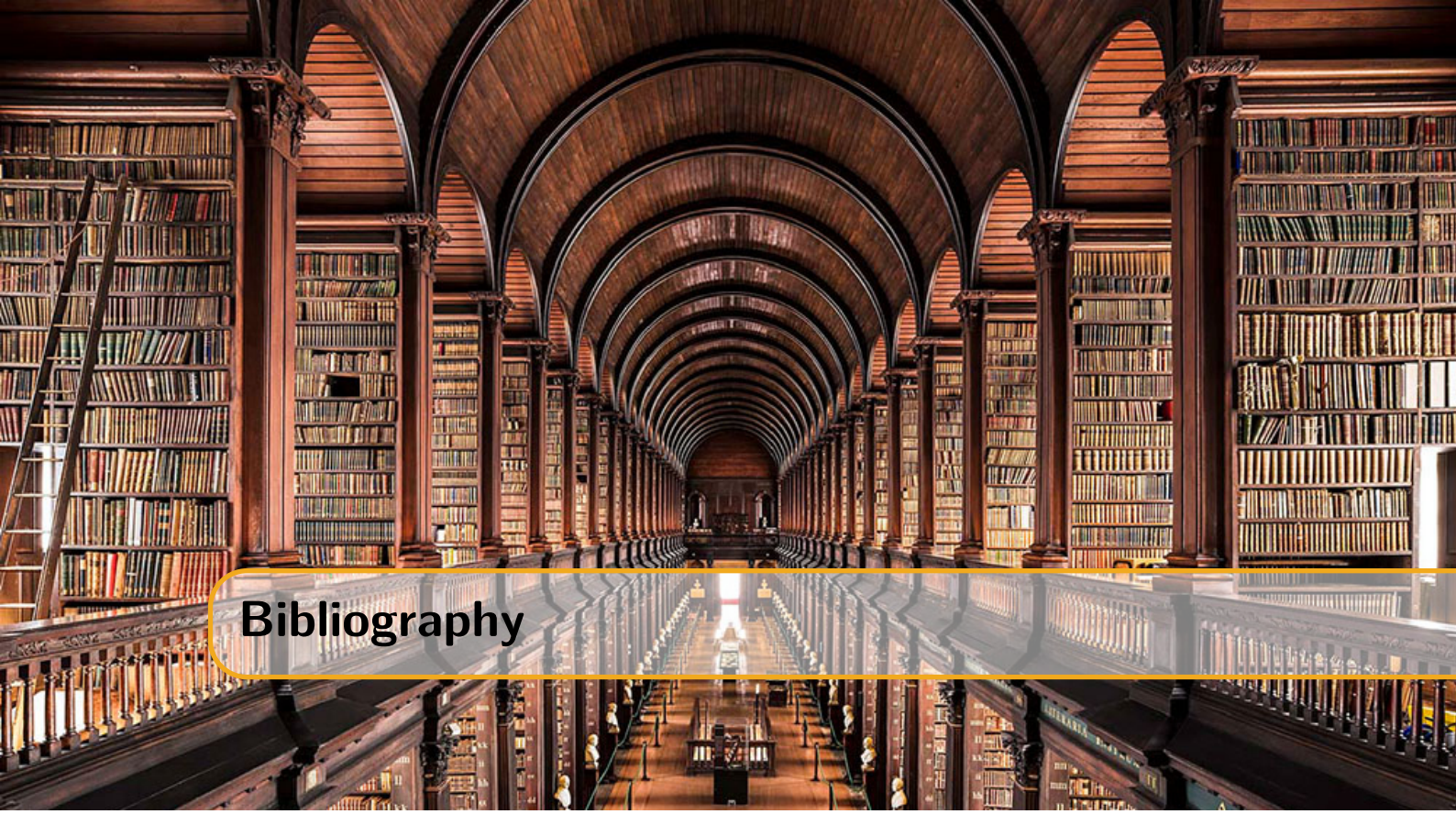
Exercise 4.1 This is a good place to ask a question to test learning progress or further cement ideas into students' minds. ■

4.7 Problems

Problem 4.1 What is the average airspeed velocity of an unladen swallow?

4.8 Vocabulary

Define a word to improve a students' vocabulary.
Vocabulary 4.1 — Word. Definition of word.



Bibliography

[1] IceCube collaboration. URL: https://docushare.icecube.wisc.edu/dsweb/Get/Document-56551/VEFproposal.pdf (cited on page 25).

[2] IceCube collaboration. URL: https://docushare.icecube.wisc.edu/dsweb/Get/Document-59303/lowup_2012_proposal.pdf (cited on page 27).

[3] IceCube collaboration. URL: https://docushare.icecube.wisc.edu/dsweb/Get/Document-59728/onlinel2proposal2012.pdf (cited on page 27).

[4] IceCube collaboration. URL: https://docushare.icecube.wisc.edu/dsweb/Get/Document-59397/DeepCoreFilterProposal.pdf (cited on page 27).

[5] IceCube collaboration. URL: https://wiki.icecube.wisc.edu/index.php/Effects_of_DOM_Mainboard_Release_443 (cited on page 29).

[6] S. Verpoest. "Search for particles with fractional charges in IceCube based on anomalous energy loss". Master's thesis. UGent, 2018 (cited on page 32).

[7] J. Künnen. "A Search for Dark Matter in the Center of the Earth with the IceCube Neutrino Detector". Master's thesis. VUB, 2015 (cited on page 34).

Index