Project Proposal

Domain Background

In this competition, I will challenge to build an algorithm that quickly reconstructs particle tracks from 3D points left in the silicon detectors.

To explore what our universe is made of, scientists at CERN are colliding protons, essentially recreating mini big bangs, and meticulously observing these collisions with intricate silicon detectors.

While orchestrating the collisions and observations is already a massive scientific accomplishment, analyzing the enormous amounts of data produced from the experiments is becoming an overwhelming challenge.

Event rates have already reached hundreds of millions of collisions per second, meaning physicists must sift through tens of petabytes of data per year. And, as the resolution of detectors improve, ever better software is needed for real-time pre-processing and filtering of the most promising events, producing even more data.

With event rates already reaching hundred of millions of collisions per second, physicists must sift through ten of petabytes of data per year. Ever better software is needed for processing and filtering the most promising events. This will allow the LHC to fulfill its rich physics programme, understanding the private life of the Higgs boson, searching for the elusive dark matter, or elucidating the dominance of matter over anti-matter in the observable Universe.

`CERN official site`

https://sites.google.com/site/trackmlparticle/

`kaggle`

https://www.kaggle.com/c/trackml-particle-identification

Problem Statement

With event rates already reaching hundred of millions of collisions per second, physicists must sift through ten of petabytes of data per year. Ever better software is needed for processing and filtering the most promising events. This will allow the LHC to fulfill its rich physics

programme, understanding the private life of the Higgs boson, searching for the elusive dark matter, or elucidating the dominance of matter over anti-matter in the observable Universe.

Datasets and Inputs

Files

The following files are available for the participants:

- sample_submission.zip: a sample submission file with score zero.
- test.zip: the test dataset with 125 events; the basis for all submissions.
- train_{1,2,3,4,5}.zip: the full training dataset with 8850 events split into 5 files for convenience.
- train_sample.zip: the first 100 events from the training dataset.
- detectors.zip: additional detector geometry information.

Dataset

A dataset comprises multiple independent events, where each event contains simulated measurements (essentially 3D points) of particles generated in a collision between proton bunches at the Large Hadron Collider at CERN. The goal of the tracking machine learning challenge is to group the recorded measurements or hits for each event into tracks, sets of hits that belong to the same initial particle. A solution must uniquely associate each hit to one track. The training dataset contains the recorded hits, their ground truth counterpart and their association to particles, and the initial parameters of those particles. The test dataset contains only the recorded hits.

https://www.kaggle.com/c/trackml-particle-identification/data

Event hits

The hits file contains the following values for each hit/entry:

```
hit_id: numerical identifier of the hit inside the event.
x, y, z: measured x, y, z position (in millimeter) of the hit in global coordinates
volume_id: numerical identifier of the detector group.
layer_id: numerical identifier of the detector layer inside the group.
module_id: numerical identifier of the detector module inside the layer.
Longitudinal view onto the detector, indicating the volume and layer numbering sche
```

The volume/layer/module id could in principle be deduced from x, y, z. They are given here to simplify detector-specific data handling.

Event truth

The truth file contains the mapping between hits and generating particles and the true particle state at each measured hit. Each entry maps one hit to one particle.

```
hit_id: numerical identifier of the hit as defined in the hits file.
particle_id: numerical identifier of the generating particle as defined in the part
tx, ty, tz true intersection point in global coordinates (in millimeters) between t
tpx, tpy, tpz true particle momentum (in GeV/c) in the global coordinate system at
weight per-hit weight used for the scoring metric; total sum of weights within one
```

Event particles

The particles files contains the following values for each particle/entry:

```
particle_id: numerical identifier of the particle inside the event.
vx, vy, vz: initial position or vertex (in millimeters) in global coordinates.
px, py, pz: initial momentum (in GeV/c) along each global axis.
q: particle charge (as multiple of the absolute electron charge).
nhits: number of hits generated by this particle.
All entries contain the generated information or ground truth.
```

Event hit cells

The cells file contains the constituent active detector cells that comprise each hit. The cells can be used to refine the hit to track association. A cell is the smallest granularity inside each detector module, much like a pixel on a screen, except that depending on the volume_id a cell can be a square or a long rectangle. It is identified by two channel identifiers that are unique within each detector module and encode the position, much like column/row numbers of a matrix. A cell can provide signal information that the detector module has recorded in addition to the position. Depending on the detector type only one of the channel identifiers is valid, e.g. for the strip detectors, and the value might have different resolution.

```
hit_id: numerical identifier of the hit as defined in the hits file.
ch0, ch1: channel identifier/coordinates unique within one module.
value: signal value information, e.g. how much charge a particle has deposited.
```

Solution Statement

For this time, we can make prediction using DBSCAN which is a type of clustering algorithm. In addition, we can measure and verify this prediction with Custom Metric.

Benchmark Model

The model with the Public Leaderboard score of 0.2078 will be used as benchmark model.

https://www.kaggle.com/mikhailhushchyn/dbscan-benchmark

Evaluation Metrics

The evaluation metric for this competition is a custom metric. In one line : it is the intersection between the reconstructed tracks and the ground truth particles, normalized to one for each event, and averaged on the events of the test set.

First, each hit is assigned a weight:

the few first (starting from the center of the detector) and last hits have a larger weight
hits from the more straight tracks (more rare, but more interesting) have a larger weight
random hits or hits from very short tracks have weight zero
the sum of the weights of all the hits of one event is 1 by construction
the hit weights are available in the truth file. They are not revealed for the test dataset
Then, the score is constructed as follows:

tracks are uniquely matched to particles by the double majority rule:
for a given track, the matching particle is the one to which the absolute majority (strictly more that 50%) of the track points belong.
the track should have the absolute majority of the points of the matching particle. If any of these constraints is not met, the score for this track is zero
the score of a surviving track is the sum of the weights of the points of the intersection between the track and the matching particle.
the score of an event is the sum of the score of all its tracks.
the final score is the average on the events of the public and private leaderboard test respectively.
A perfect algorithm will have a score of 1, while a random one will have a score 0. An example implementation can be found in the trackml python library.

https://www.kaggle.com/c/trackml-particle-identification#evaluation

https://github.com/LAL/trackml-library

Project Design

`work flow`

- input particle physics
- deep understanding of the problem
- EDA(understanding of datasets)
- Investigation of method
- Implementation and validation

input particle physics

I do not have any knowledge of particle physics at all, so I need to do inputs.

https://www.kaggle.com/c/trackml-particle-identification/discussion/55726

deep understanding of the problem

Based on that knowledge I will understand this issue again.

EDA

Discovering a data set has a major impact on model selection and feature engineering. I can get a lot of knowledge from EDA.

Investigation of method

This different color problem is different from regular Kaggle contest. Based on knowledge of the domain, it is necessary to investigate many methods. DBScan is one of the new learning methods.

Implementation and validation

After investigation, I implement and verify various methods and build final script.