Project Proposal

Domain Background and Problem Statement

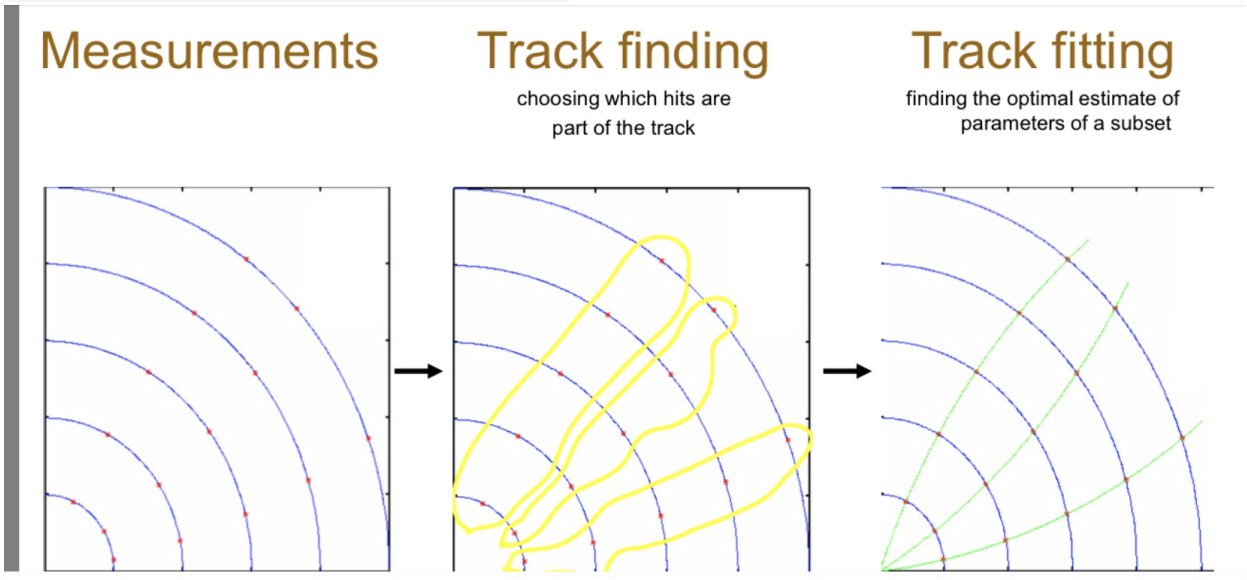
At the laboratory called CERN, we are doing research on proton collision to elucidate the origins of the universe.
However, in this proton study, a large amount of data is generated and physicists have to process about 10 petabytes of data per year.

When elementary particles collide with a detector, coordinates in three dimensions can be obtained. Actually this coordinates physicist need to figure out the trajectory of the particle.

In other words, it is necessary to join the trajectories related to certain particles based on the large amount of coordinate data obtained.

However, it is impossible to join the orbits actually by hand, as there is huge data.So we use machine learning to predict this trajectory. Physicists can concentrate on their duties by predicting orbit.

`Predict trajectory by machine learning`



`CERN official site`

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

`kaggle`

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

Datasets and Inputs

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.

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

These data sets contain the following information:

- Event hits
- Event truth
- Event particles
- Event hit cells

The data is divided into the above data, and when all the data are combined, there are 11 characteristic quantities. It also consists of 8850 rows. This data can be handled by a dedicated library trackML.

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

task

To predict the trajectory of certain elementary particles. Although the data set contains data on the trajectories of many elementary particles, it is unknown which elementary particle the data is on in the orbit of some elementary particle.

output

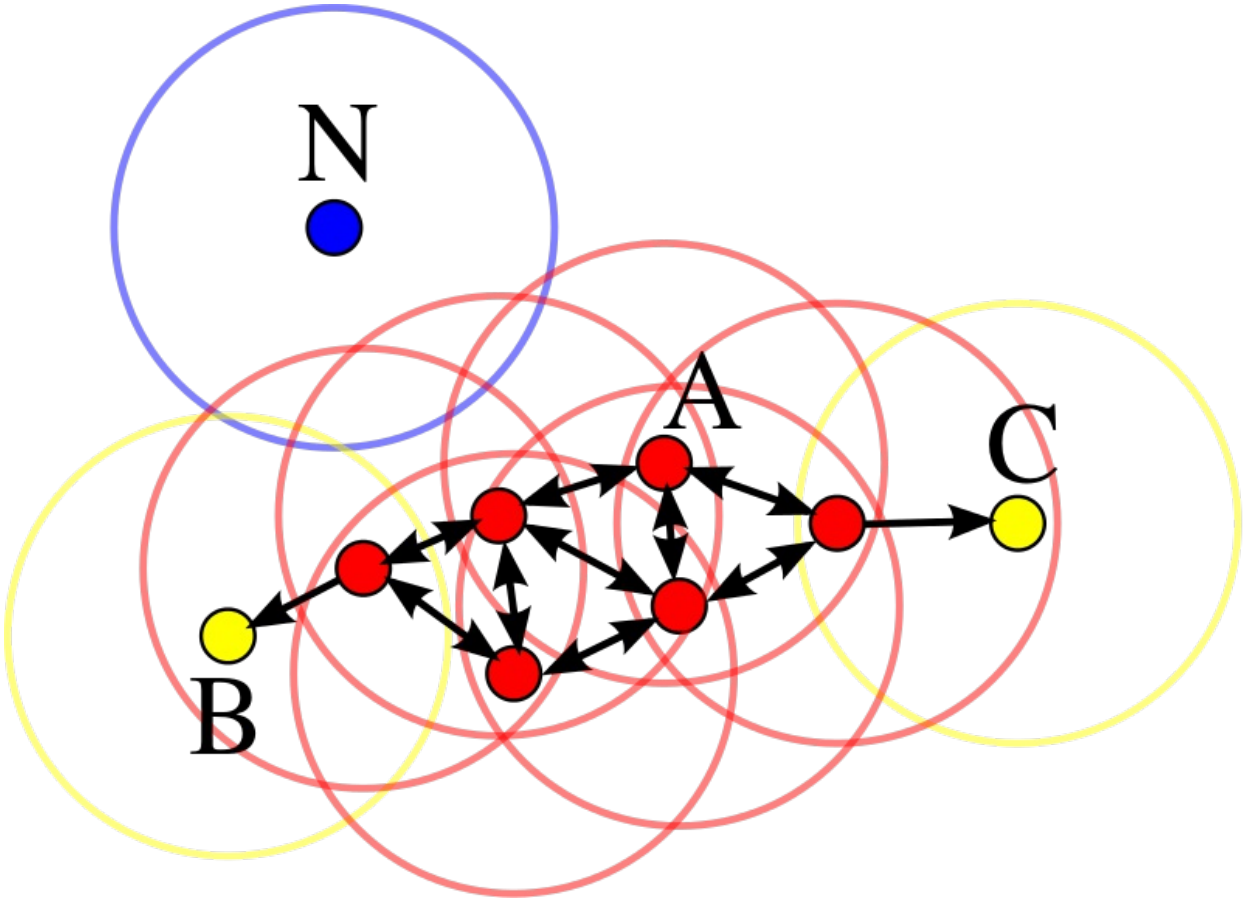
Identify which elementary particle trajectory the input data belongs to (Classification problem)

Solution Statement

For this time, we can make prediction using DBSCAN which is a type of clustering algorithm. By using the clustering algorithm, we can predict which orbit the data belongs to.

In addition, we can measure and verify this prediction with Custom Metric.

`detail`



DBSCAN is a density-based clustering algorithm. As shown in the illustration, it is expected that the density of each particle will be smaller if it is the same particle trajectory. DBSCAN can cluster in multidimensional without specifying how many clustering is done beforehand.

https://arxiv.org/pdf/1012.6009.pdf

Benchmark Model

I used the knn approach to check if the problem can be solved.The model with the local score of 0.09900 will be used as benchmark model.

`The following code is forked from the kernel`
https://www.kaggle.com/lasershow/knn-approach/code

Evaluation Metrics

`summary`
The evaluation metric for this competition is a custom metric.
Each hit is weighted and the score is calculated as follows.
track is uniquely matched to particles by double majority rule. Match and sum the remaining

tracks and make it a total of events.

`detail`
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.

`justification`
In this contest, we need to use dedicated metrics in elementary particle physics, not metrics that are normally used to properly evaluate problems in elementary particle physics.

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

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

Project Design

`work flow`

- input particle physics approach
- EDA(understanding of datasets)
- Preprocessing
- Implementation and validation

input particle physics approach

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

Especially, it is necessary to input the approach about this problem.For example, the following method is one of the methods discussed this time.

- Conformal Mapping
- Hough Transformation
- Track Following based on the Kalman Filter
- Cellular Automaton with the Kalman Filter for fitting

EDA

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

https://www.kaggle.com/wesamelshamy/trackml-problem-explanation-and-data-exploration

For example, by conducting EDA as described above, many knowledge can be obtained.

- What is the distribution of data?
- What kind of data is it?

Preprocessing

`feature engineering`
We know that the particle's trajectory is almost spiral.
So, there is an idea to emphasize this point as preprocessing.

$$r_1 = \sqrt{x^2 + y^2 + z^2}$$

$$x_2 = x/r_1$$
$$y_2 = y/r_1$$

$$r_2 = \sqrt{x^2 + y^2}$$

$$z_2 = z/r_2$$

※ x_2,y_2,z_2 are new features.

`Standardization`

The x, z, y coordinates take large values. (1500 to -1500) Therefore, I will standardize.

Implementation and validation

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

Methods as described above,

- Conformal Mapping
- Hough Transformation
- Track Following based on the Kalman Filter
- Cellular Automaton with the Kalman Filter for fitting

Verify using a deep learning approach.

- 3DCNN
- LSTMC

tuning

Since the DBSCAN algorithm is used in many of the contests, tuning on the DBSCAN algorithm will be described.

The most important parameter is `eps`. `eps` represents the maximum distance between two samples and determines if this distance is the same category.

http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html