High energy learning for a better particle tracking

TrackML Particle Tracking Challenge

Andrea Lonza
TS, Italy
andrea.lonza@gmail.com

A2.

Competition Name: TrackML
Team Name: Andrea Lonza
Private Leaderboard Score: 0.75845
Private Leaderboard Place: 11

A3. Summary

The overall algorithm can be seen as two distinctive parts: the first predict the tracks by clustering the hits with the same properties, the second extend the start and the end of these tracks (a sort of track fitting algorithm).
The first part adopts unsupervised algorithms and the second one supervised algorithms.

To cluster together the hits of the same track, the hits have been converted in polar coordinates and rotated by an angle proportional to their distance from the center. This unrolling effect has been done with multiple angles. Once the hits of the same track lie nearby the density-based clustering algorithm can correctly cluster them together.
The clustering algorithm takes as input 5 features that remain constant across the hits of the same track.
Because the originating point of the tracks isn't always (0,0,0), has been used the z shifting techniques (described in the discussions forum)

The extension of the tracks is done by collecting the hits that lie nearby the start and the end of a track (in a 2D space) and using a supervised classification algorithm to predict whether these hits belong to the track.

The main libraries used are pandas, numpy, lightgbm and scikit_learn.

To train the supervised model, at least 4GB of RAM is required. This model took about 2h to be trained on a high mid-range CPU (i5-6267).

A4 & A5. Description

Because the most original part of this approach is the supervised track extension part, in this section I will concentrate on it, but before it, I will explain briefly some very interesting tricks that I used in

the clustering part of the algorithm.

Track clustering

- **Merge: Choose the track with the higher number of layers** – if for a given hit two or more tracks are founded, differently on how has been done by other teams, a hit will be assigned to the track that goes through the higher number of layers. This strategy has been adopted because only in some rare case two hits of the same track are in the same layer (or module). This mechanism will choose the best possible track and support the tracks that extend in more layers.
- **Ensembling stategy** - Many tracks obtained with a different Z shift, unroll parameters and clustering hyperparameters have been ensembled. The ensemble strategy (multiple "merge" stages) will boost the most precise tracks (with lower eps - local radius for expanding clusters)

Tracks extension

The innovation brought by this approach is to refine the tracks predicted by the clustering algorithm, extending the tracks using a classification algorithm.

Because the first phase (i.e. the unsupervised one) isn't able to cluster correctly most of the first and last hits of the tracks, the track extension algorithm is a fundamental part of the approach capable of increasing dramatically the score.

The hits in Cartesian coordinates are converted to polar coordinates r and φ where $r=\sqrt{x^2+y^2}, \phi=arctan2(y,x)$. Then for each track are collected at most N hits that lie nearby the first and last hit of the track.
Then, the hits that have been collected pass through a filter (to discard the improbable ones by handcraft rules) and for each of the remaining ones about 60 features are created. These features take into consideration the polar and cartesian coordinates of the hits and their position with respect to the hits of the track.

These features are then given to a LightGBM classification algorithm (high performance gradient boosting framework based on decision tree) that will output the probability to belong to the track. If the probability reachs a given threshold, the hit will be added to the track. This procedure is repeated 2 times.
To train the LightGBM model, has been created a dataset in the same way just described, with the only exception that it's checked in the truth file if the hit actually belongs to the track.

Because of the limited availability of computational power during the competition, the dataset has been collected using only 10 events. So, overall the dataset is constituted by 10 Million pair of features-truth_value where the features encode the information of the track and the proposed hit, and truth_value contain 1 is the hit belongs to the track, 0 otherwise.

The extension algorithm increases the performance from 0.68x to 0.76x in only 15 minutes.

A6. Interesting findings

- In the clustering algorithm, the most important tricks are the two described in the A4 section, namely the use of the merge technique that takes the track with a higher number of hits from different layers and the ensembling strategy.

- The extension track algorithm was the “feature” that allowed to boost incredibly the final performance (from 0.68x to 0.76x)

A7. Simple Features and Methods

- Using a less aggressive ensembling approach, the final score will decrease to about 0.74 but with an overall computational time reduction of about one half

A8. Model Training and Execution Time

- To create the dataset and train the supervised model took about 2H on a 4 core i5-6267 CPU and with 4GB of RAM. The dataset can be created with parallelization techniques to further increase the speed. Furthermore, the LighGBM algorithm can be run on GPU, which can speed up the computations.
- As language it's been used python
- To run the supervised and unsupervised models for predictions, it takes about 1:20h on a single core with a i5-6267 CPU but on a 48 cores high-end CPU the models run in 6H for all 125 test events (using parallelization techniques). Specifically, about 5H for the ensemble clustering algorithms and 1H for the supervised track extension.
- The simplified version could take about 3H for predict 125 events on the 48 cores CPU (using parallelization techniques).

A9. Outlook

- With more time and more computational power, more and better features can be used on the DBSCAN algorithm and can be gathered a bigger dataset to train the supervised algorithm.
- The track extension can be applied more than two times to reach better performance and can be adapted as track-fitting algorithm