Machine learning applied to EM showers of the OPERA experiment

Vladislav Belavin, Andrey Ustyuzhanin

Higher School of Economics

June 1, 2018

SCHOOL OF DATA ANALYSIS

SCHOOL OF DATA ANALYSIS

Introduction

Problem statement

Problem: identify all showers in the ECC(Emulsion Cloud Chamber) brick and for each shower reconstruct its energy, initial position and direction.

Setup: $\sim 3 - 5 \cdot 10^6$ background basetracks and 50-200 showers in one ECC brick.

Metrics:

Energy resolution:	$\Delta E / E$
Quality of initial position reconstruction:	MAE over $x_{start}, y_{start}, z_{start}$
Quality of direction reconstruction:	MAE over θ_x, θ_y
Ratio of recovered showers:	recovered showers / total showers
Track classification(indirect metric):	ROC-AUC and PR-AUC

SCHOOL OF DATA ANALYSIS

Data

Each basetrack from ECC brick is described by 6 variables: $x, y, z, \theta_x, \theta_y, \chi^2$, i.e. position, direction and fit quality.

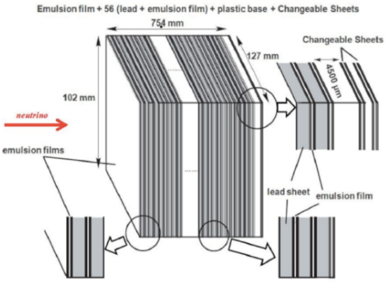
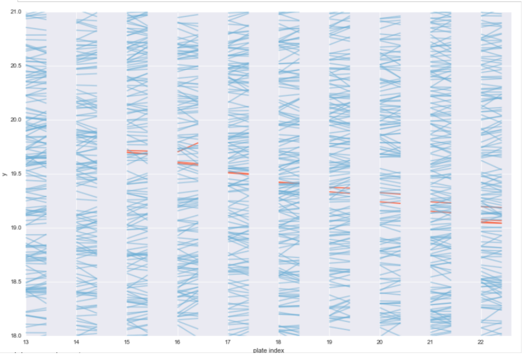
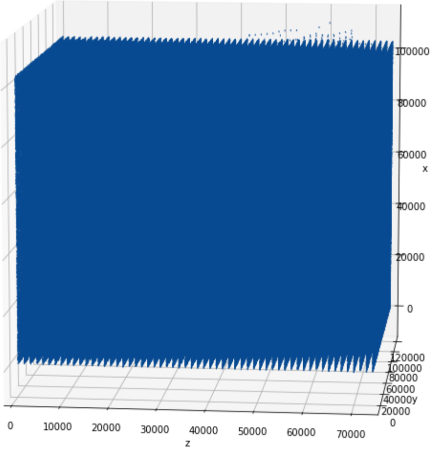
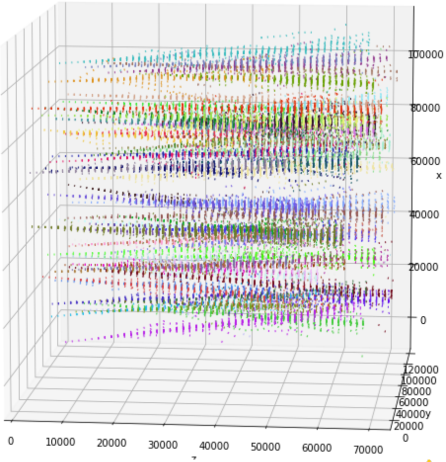


Figure 2.4 – Schematic structure of an ECC brick.



Brick example.





SCHOOL OF DATA ANALYSIS

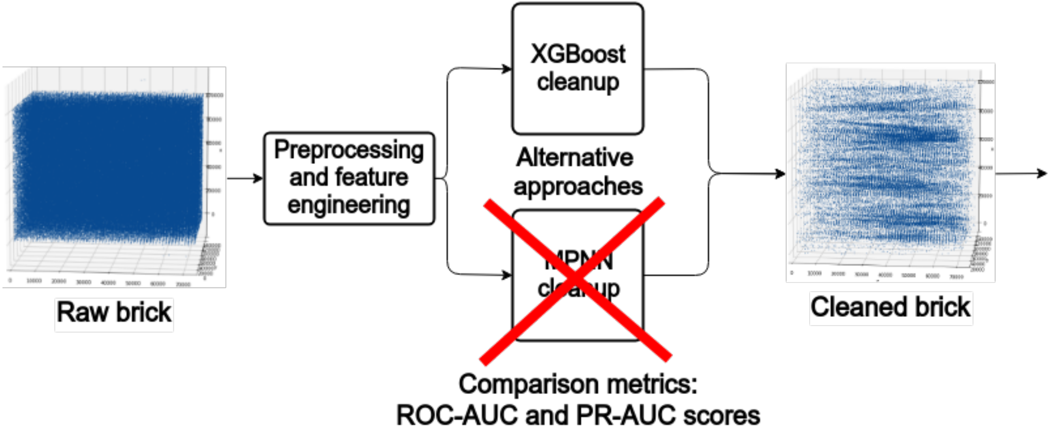
Previous works

- ‘Search for Tau Neutrinos in the $\tau \to e$ Decay Channel in the OPERA Experiment’ B. Hosseini
  - only one shower with **known** origin;
  - $\sigma(\Delta E/E) = 0.20$
- ‘Machine-Learning techniques for electro-magnetic showers identification in OPERA datasets’ A.Ustyuzhanin, S. Shirobokov, V. Belavin, A. Filatov
  - only one shower with **unknown** origin;
  - $\sigma(\Delta E/E) \sim 0.27$

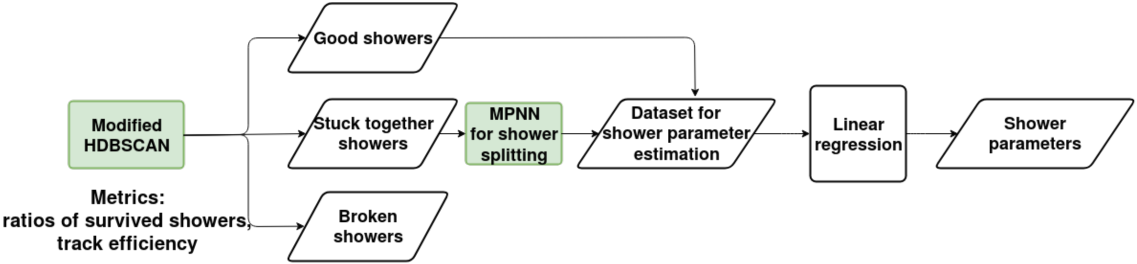
SCHOOL OF DATA ANALYSIS

Methodology

Pipeline


Raw brick
Preprocessing and feature engineering
XGBoost cleanup
Alternative approaches
MPNN cleanup
Cleaned brick
Comparison metrics:
ROC-AUC and PR-AUC scores

Pipeline (continued)



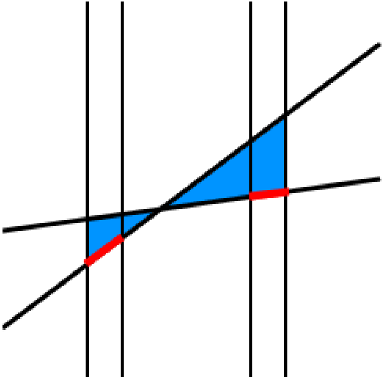
SCHOOL OF DATA ANALYSIS

Cleanup

Preprocessing

In preprocessing I am building graph with nodes represented by basetracks. Two nodes(basetracks) are connected with the edge if and only if:

- between them less than 3 layers;
- one of the basetracks lies in the cone of 16 (mrad) with origin in another basetrack;
- integrated distance(blue area on the pic) between tracks less than threshold.



SCHOOL OF DATA ANALYSIS

Feature engineering and XGBoost cleanup

For cleaning XGBoost is applied.

For each basetrack 63 features have been constructed. 3 original basetrack features:

- $\theta_{x,y}$ and χ^2

60 pairwise basetrack features for 4 nearest(with lowest integrated distance) neighbors:

- $\Delta\theta_{x,y}$ and Δx, Δy, Δz between tracks;
- $IP_{x,y}$ for both tracks;
- integrated distance;
- $\theta_{x,y}$, χ^2 of neighbor;
- energy-like feature and likelihood induced from multiple scattering theory(details in backup slide).

SCHOOL OF DATA ANALYSIS

XGBoost cleanup

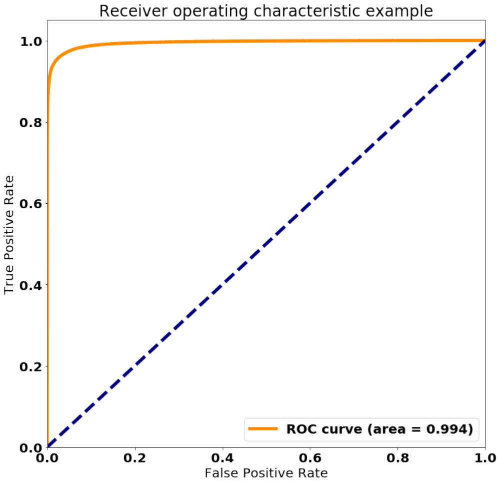


Figure 1: ROC-AUC=0.994

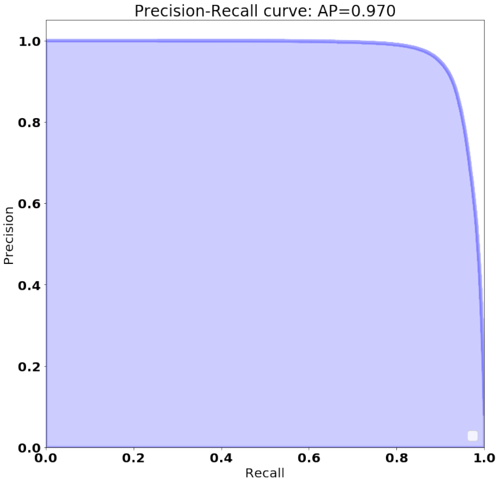


Figure 2: PR-AUC=0.97

Clusterization

Modified HDBSCAN. Algorithm

Step 1. Calculate all pairwise distances.

I.e. calculate all $d(i, j)$. In our case, $d(\mathrm{basetrack}_i, \mathrm{basetrack}_j) =$ integrated distance from preprocessing step;

Step 2. Define mutual reachability distance with parameter 'k'.

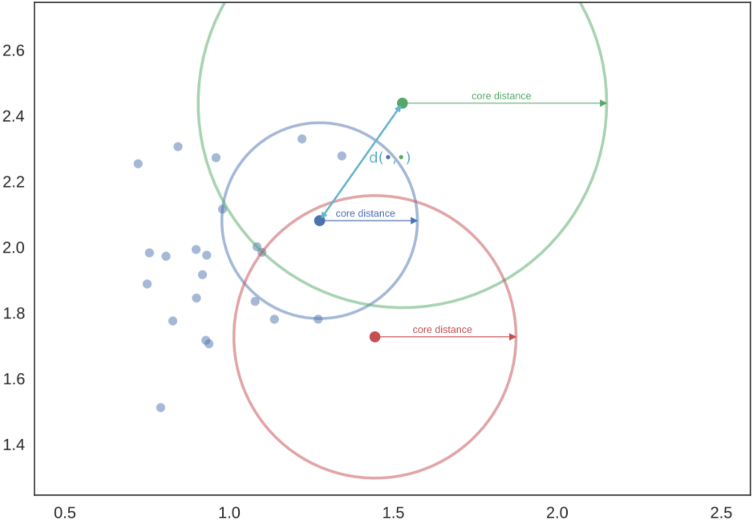
$d_{mreach-k} = \max\{core_k(\mathrm{basetrack}_i), core_k(\mathrm{basetrack}_j), d(\mathrm{basetrack}_i, \mathrm{basetrack}_j)\}$.

Where $core_k(\mathrm{basetrack}_i)$ is a distance from $\mathrm{basetrack}_i$ to the k-nearest neighbor.

At this point you have complete graph of distances.

SCHOOL OF DATA ANALYSIS

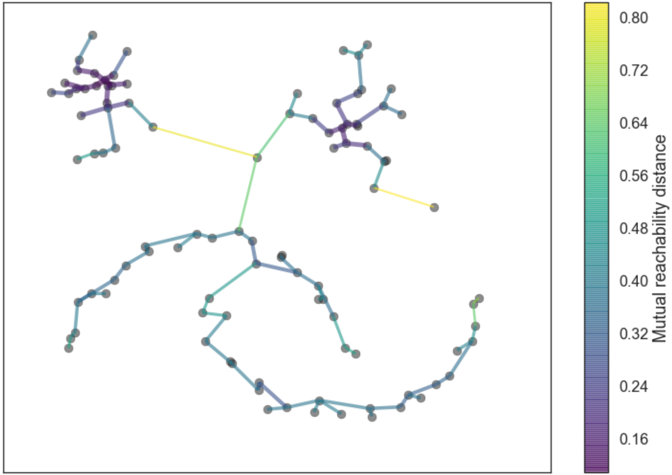
Modified HDBSCAN. Algorithm



SCHOOL OF DATA ANALYSIS

Modified HDBSCAN. Algorithm

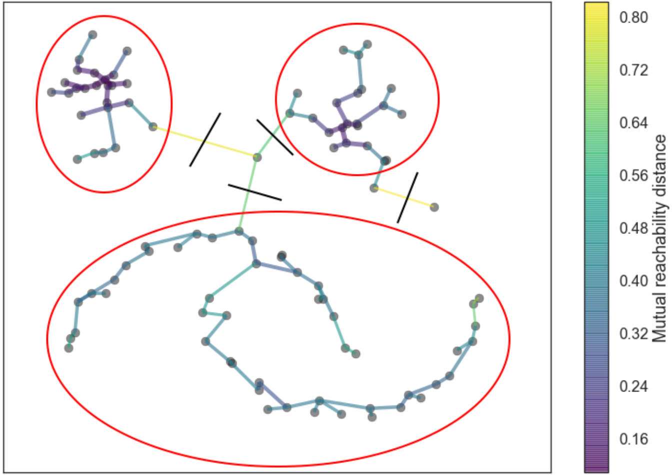
Step 3. Construct minimum spanning tree, i.e. tree with the lowest sum of edges weights.



SCHOOL OF DATA ANALYSIS

Modified HDBSCAN. Algorithm

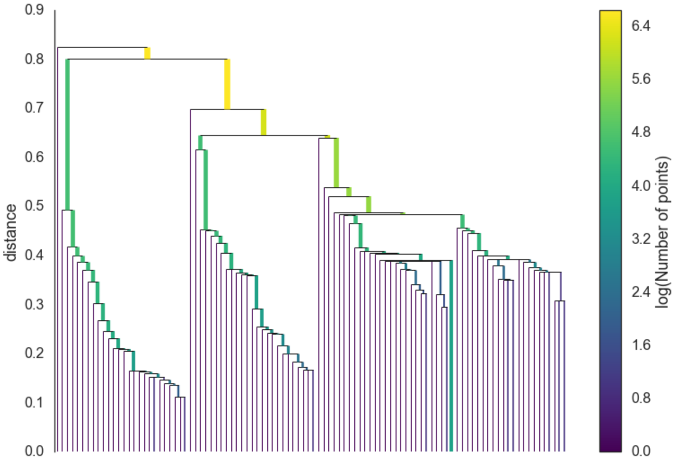
Step 4. Iteratively delete edges with the highest weight.



SCHOOL OF DATA ANALYSIS

Modified HDBSCAN. Algorithm

Step 5. Choose cut level.



SCHOOL OF DATA ANALYSIS

Modified HDBSCAN for shower extraction. Results.

Modification to original algorithm:

- physically motivated stop criteria;
- modified core distance calculation.

Number of showers in brick	50	100	150	200
Ratio of survived showers	93 %	93 %	91 %	88 %
Ratio of stuck showers	3 %	3 %	3 %	3 %
Ratio of broken showers	1 %	0 %	1 %	2 %
Ratio of lost showers	3 %	4%	5 %	7 %

SCHOOL OF DATA ANALYSIS

Showers separation with MPNN

Graph terminology

- $V = \{v_i\}_{i=0}^{N}$ – set of vertices with:
  - associated features: $x_i \in \mathbb{R}^{D_1}$
  - hidden state: $h_i \in \mathbb{R}^{H_1}$
- $E \subset 2^V \; \& \; \forall \mathrm{e} \in \mathrm{e} \in E \; (|\mathrm{e}| = 2)$ – set of edges with:
  - associated features: $x_{ij} \in \mathbb{R}^{D_1}$
  - hidden state: $h_{ij} \in \mathbb{R}^{H_1}$
- $G(V, E) = \langle V; E \rangle$ – graph;

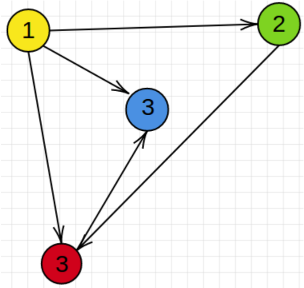
SCHOOL OF DATA ANALYSIS

MPNN

MPNN stands for **Message Passing Neural Networks**.

The core idea of MPNN is passing 'messages' from node to node and updating states of nodes accordingly to received 'messages', thus allowing taking into account both structural-wise and node-wise information.

There could be several steps of passing 'messages' and nodes updates. To distinguish them we are going to use upper-score symbol t, $t \in \{1, 2, ..., T\}$



SCHOOL OF DATA ANALYSIS

MPNN(continued)

So, in a nutshell to define MPNN one have to define three operations:

- message passing: $m_{ij}^t = f_{ij}^t(h_i^t, h_j^t, h_{ij}^t)$;
- message aggregation: $M_j^t = g_j^t\left(\{m_{ij}^t\}_i\right)$;
- node update: $h_i^{t+1} = U_i^t(M_j^t, h_i^t)$.

Usually you also want to define readout function to make predictions:

- over single node: $R_i(h_i^T)$;
- or over whole graph: $R_i\left(\{h_i^T\}_i\right)$.

All functions could be handcrafted or parametrized by neural network. In latter case it is possible to learn parameters through backpropagation.

SCHOOL OF DATA ANALYSIS

MPNN for shower separation

To train MPNN for shower splitting I compare two loss functions.

Traditional **siamese loss** or **contrastive loss**:

$$L(x_i, x_j) = (1 - Y)D_W^2(x_i, x_j) + Y \max(0, m - D_W(x_i, x_j))^2$$

$$D_W(x_i, x_j) = ||x_i - x_j||_2$$

Y = 0 when both tracks belong to the same shower and Y = 1 otherwise.

After training this model we can use it to split stuck together showers or as an additional distance for HDBSCAN algorithm.

SCHOOL OF DATA ANALYSIS

MPNN for shower separation(continued)

Batch centered loss:

- For each class robustly find its center: $\mathrm{center}_i = \mathrm{median}(x_i)$
- similarity loss: $L_{similarity} = (\mathrm{center}_i - x_i)^2$
- contrastive loss: $L_{contrast} = \max(0, m - |\mathrm{center}_i - \mathrm{center}_j|)^2$

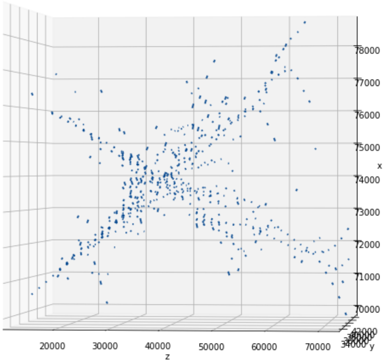
Advantages in comparison with siamese loss:

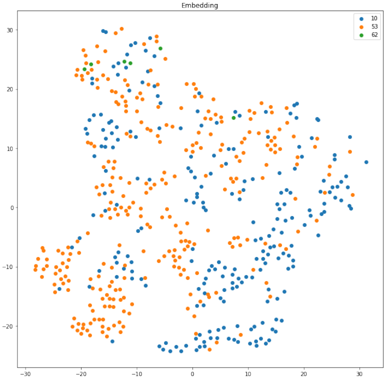
- $\sim O(N)$ operations;
- no need for sophisticated sampling strategies;
- more robust to outliers(in theory).

SCHOOL OF DATA ANALYSIS

MPNN for shower separation(continued)

Experiments with MPNN still in progress due to computational complexity of MPNNs on big graphs.





SCHOOL OF DATA ANALYSIS

Shower parameters estimation

Parameter estimation with linear models

Features:

- number of selected basetracks;
- spatial parameters of shower, i.e. statistics over $x, y, z, \theta_x, \theta_y$.

For estimation we use Theil-Sen linear estimator which is more robust to outliers than least square estimator.

SCHOOL OF DATA ANALYSIS

Parameter estimation with linear models

Result on test data for different number of showers in brick:

Number of showers in brick	50	100	150	200
$\sigma\left(\frac{\Delta E}{E}\right)$	0.28	0.27	0.28	0.28
MAE(x, y)	0.18 (mm)	0.23 (mm)	0.23(mm)	0.22 (mm)
MAE(z)	1(mm)	1.2(mm)	1 (mm)	1.1 (mm)
MAE(tx, ty)	13(mrad)	14(mrad)	15(mrad)	13(mrad)

SCHOOL OF DATA ANALYSIS

Summary

- works for $\sim 50 - 200$ showers in a brick;
- no a priori information;
- $\sim 90\%$ of showers recovered;

Future plans:

- optimize MPNN for large graphs;
- try unsupervised losses for MPNN;
- generalize HDBSCAN on directed graphs;
- try PointNet/PointNet++(NNs for point cloud segmentation) on this task.

SCHOOL OF DATA ANALYSIS

Backup

Energy-like feature

From Molier's theory of multiple scattering one can derive following probability distributions on track directions and positions:

$$P(z,\theta) = \frac{2\theta}{\langle\theta^2\rangle} \exp\left(-\frac{\theta^2}{\langle\theta^2\rangle}\right), \langle\theta^2\rangle = \theta_s^2 z = \left(\frac{E_s}{\beta cp}\right)^2 \frac{z}{X_0}$$

$$Q(z,\theta_x) = \frac{1}{\sqrt{2\pi}\sigma_{\theta_x}} \exp\left(-\frac{\theta_x^2}{2\sigma_{\theta_x}^2}\right), \sigma_{\theta_x}^2 = \frac{\langle\theta^2\rangle}{2}$$

$$S(z,x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{\theta_x^2}{2\sigma_x^2}\right), \sigma_x^2 = \frac{\theta_s^2 z^3}{6}$$

Using this distributions we can estimate energy($E \approx cp$) for all pairs of tracks and it's likelihood.

SCHOOL OF DATA ANALYSIS