Machine-Learning techniques for electro-magnetic showers identification in OPERA datasets

Vladislav Belavin[1,3], Artem Filatov [1,2], Sergey Shirobokov[1,2] and Andrey Ustyuzhanin[1,2,4]

[1] Yandex School of Data Analysis, [2] NRU Higher School of Economics, [3] MIPT, [4] INFN Napoli

Introduction

We have investigated different approaches to the recognition of electromagnetic showers in the data which was collected by the international collaboration OPERA. The experiment was initially designed to detect neutrino oscillations, but the collected data can also be used for the development of the machine learning techniques for electromagnetic shower detection in photo emulsion films. Such showers can be used as a signal of the Dark Matter interaction.

Problem statement

OPERA detector consists of volume elements (bricks). Each brick contains 56 sequential emulsion films interleaved with lead. Each layer has particle tracks. Coordinates of each track are known.

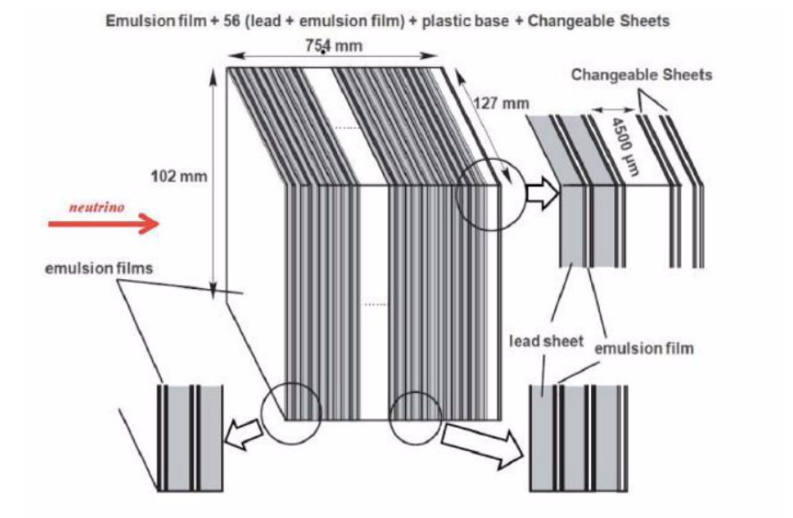

Figure 1: Brick schematic structure

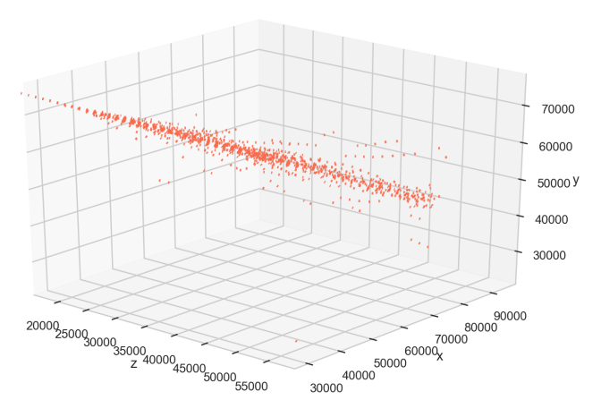

Figure 2: EM-Shower example

Each brick contains about 2.4M background events and about 300-500 signal tracks per brick. Background tracks were subsampled from last 10 layers of a real brick. Electromagnetic shower (EM-shower) tracks were generated by Monte-Carlo. Each brick contains a single shower.

The goal is to design an algorithm that can identify signal tracks of electromagnetic-showers inside the brick volume.

Baseline solution

The algorithm developed by OPERA [2] uses a priori information of the EM-shower origin. The cone with opening angle 50 mrad in direction of electron was considered. For selected tracks impact parameter, angle difference and $\chi2$ were taken as features. Boosted decision trees were fitted. The resulted ROC AUC curve and energy dependency graph are below.

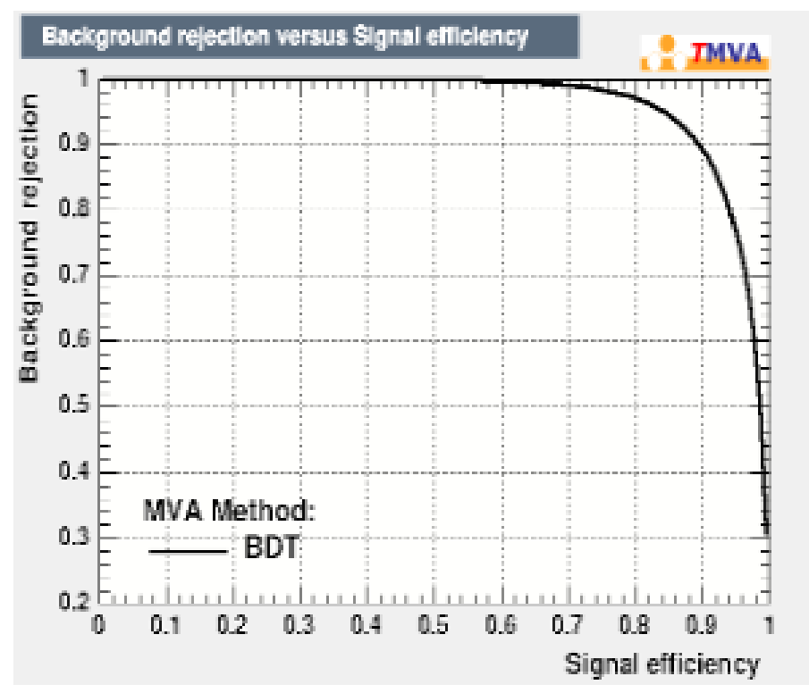

Figure 3: ROC AUC

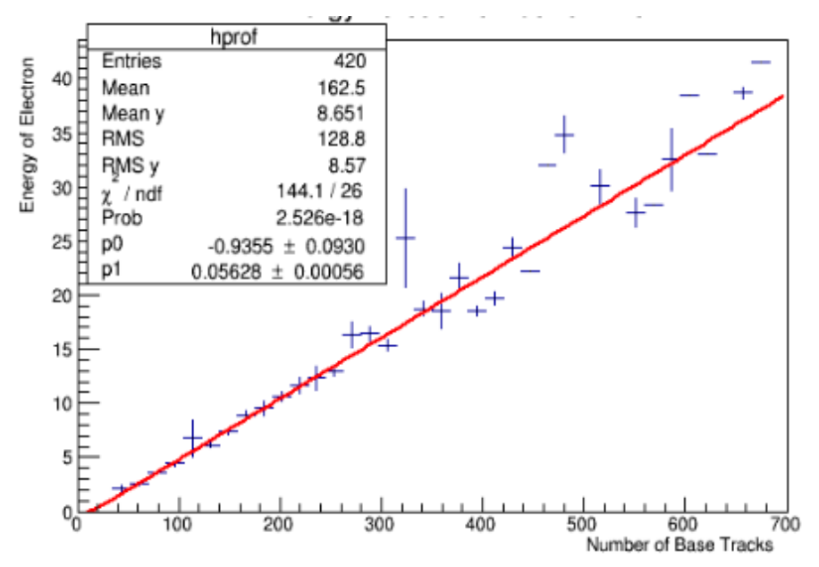

Figure 4: Energy dependency

The resulted energy resolution was 0.207.

Artem's solution

The algorithm proceeds in two steps. At the first step, we clear the brick using SVM saving only sequential structures. At the second step, we use CRF to classify sequences of noise and signal tracks.

```
Algorithm 1: SVM step
for every layer in the brick do
    for every track in a layer do
        Find neighbors on the next layer;
        Find probability of each neighbor to be a continuation of the track using SVM;
        if highest probability is bigger than threshold or track has more than h ancestors then
            Leave track;
        else
            Delete track;
```

We apply CRF to the dual graph of the original sequence and features are computed pairwise. The structure of the CRF looks following.

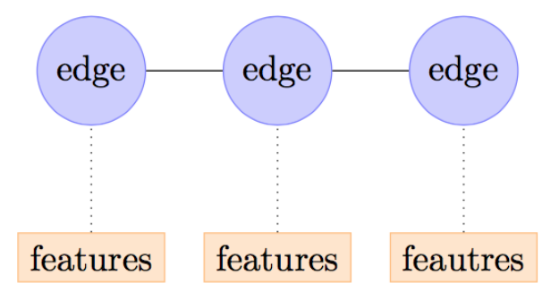

Figure 5: Chain CRF

At the both steps we use the same set of features.

1. Impact parameters to both direction
2. Euclidean distance between tracks
3. Tangent of angle
4. Difference in projection angles to X and Y
5. Chi2 for both tracks

On the pictures below you can see the cleared brick after SVM and CRF steps respectively.

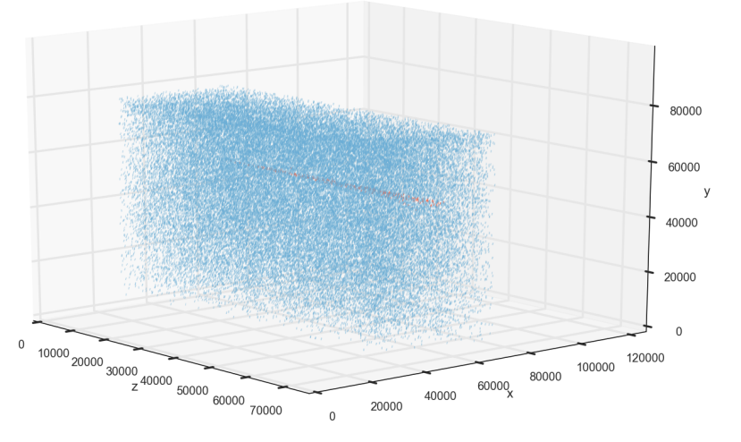

Figure 6: SVM step.

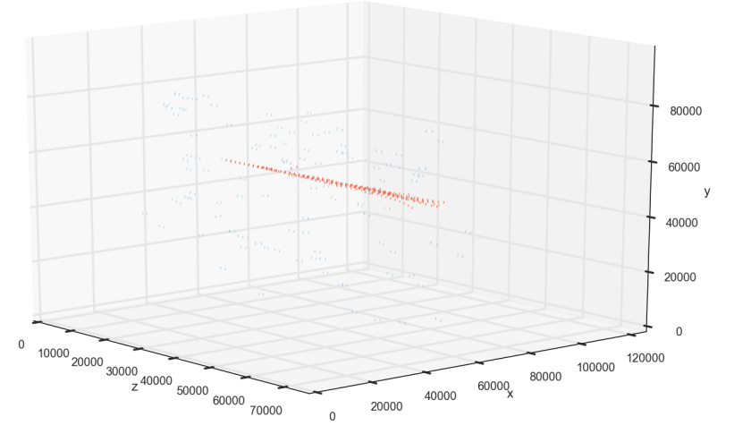

Figure 7: CRF step.

The SVM step can be applied multiple times to facilitate the work for the CRF.

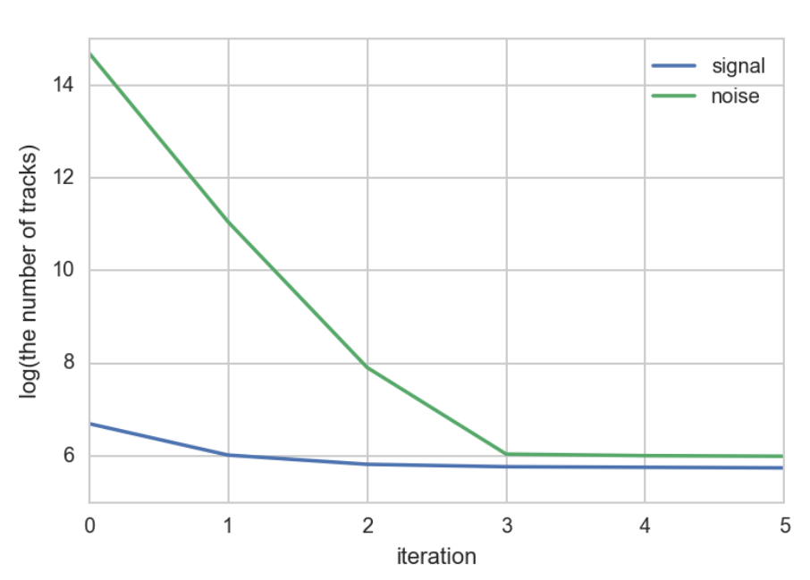

Figure 8: Dynamic of signal and noise in the SVM step.

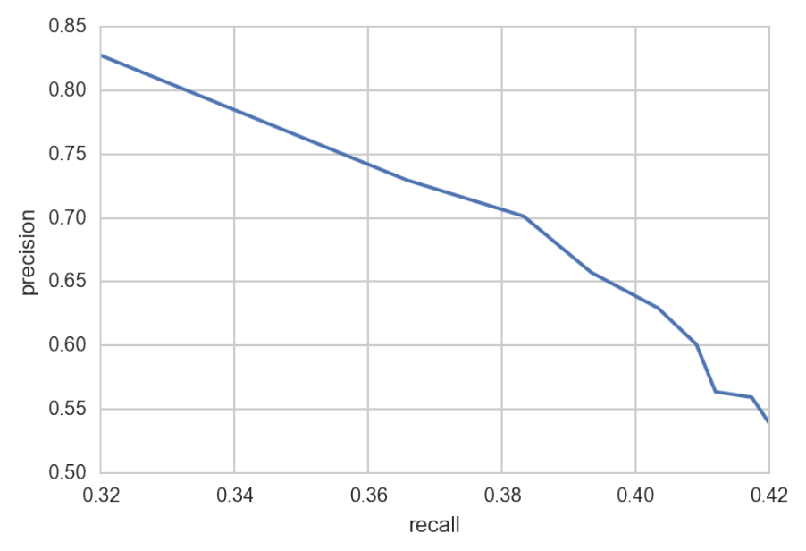

Figure 9: Precision/recall curve for SVM step (Full brick on last iteration).

The mean precision/recall (inside the cone) for the algorithm is about 0.48/0.9. Energy resolution is 0.43 for events where energy is lower 20 GeV.

Sergey's solution

1. Each layer of emulsion is considered with it's two neighbor layers. For each track a projection of it's direction is obtained for both layers and features calculated for 10 nearest tracks. The CatBoost [1] classifier is applied on given features.

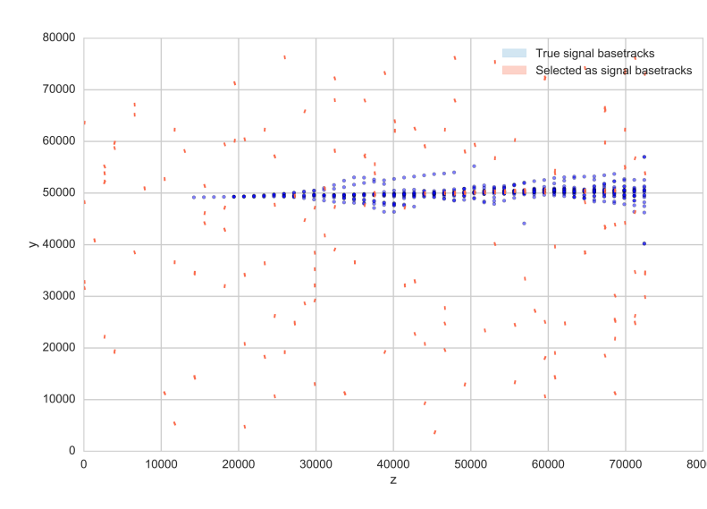

Figure 10: Red points are selected as signal. Blue is the true signal.

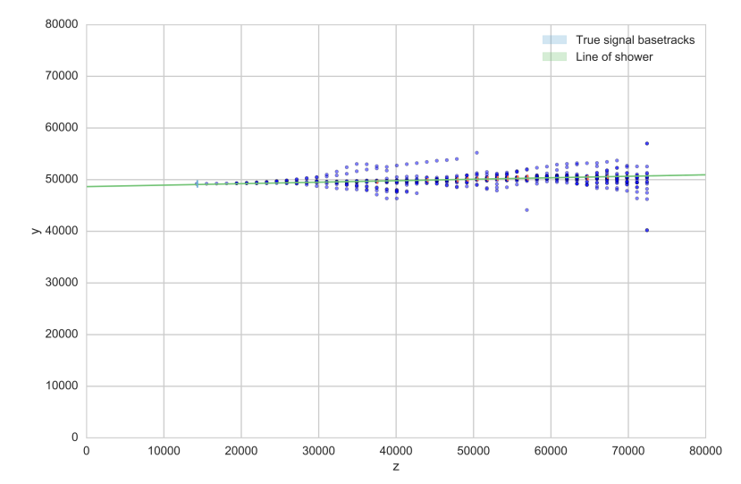

Figure 11: Red points are selected as signal. Blue is the true signal. Green is shower line

2. On this stage of the algorithm provides topological filter, by deleting all unclustered tracks. To do this, the median of X, Y coordinates in each layer of brick are calculated. Calculating centers of selected tracks in each layer, and fitting PCA on them, selecting first component, one gets the direction of the shower.

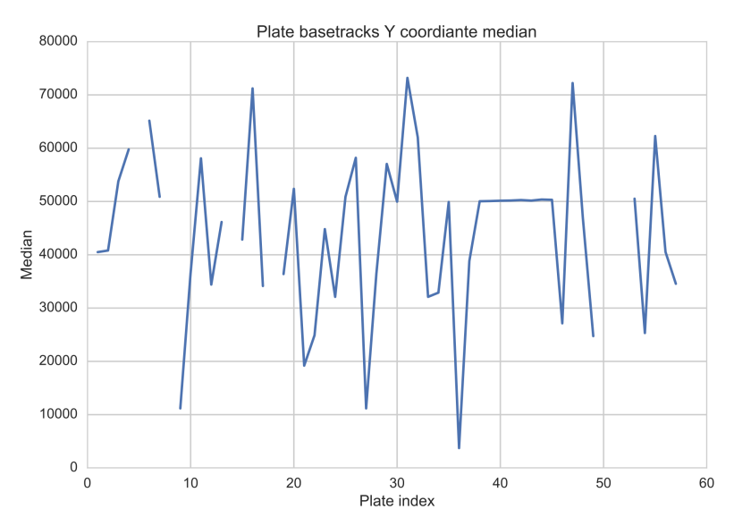

Figure 12: Y Median values by plates after first stage.

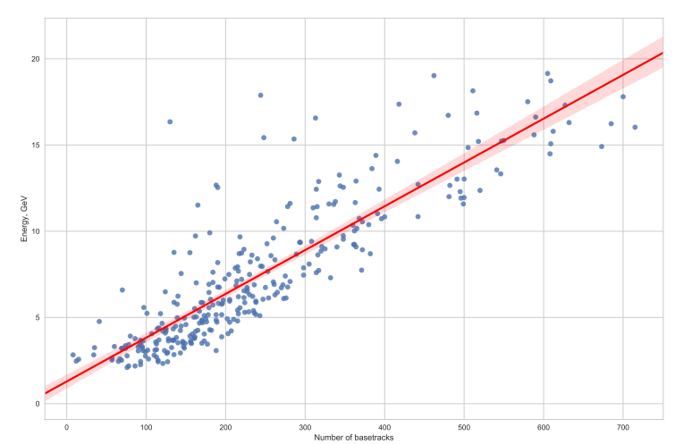

Figure 13: True energy from recon number of tracks. Red is fitted regression.

3. On each iteration it is assumed that we know tracks at plate number N. Then, building line on known tracks, one finds intersection point of line and plates N-2 and N-1. Now, algorithm supposes that shower origin is at plate N-2, find tracks in small region near the point at plate N-1 and calculate Impact Parameter to "supposed" origin.
4. Algorithm was run on 400 events and 350 events with more than 200 tracks were selected. For them, mean distance error in XY plane is 0.4 mm, and mean distance error in Z coordinate 2.4 mm. About 85% of showers origins detected within 2 plates distance from real origin. After direction and initial point of shower are approximately found, the used OPERA algorithm [2] is applied. Mean average precision is 0.81 ± 0.07.
Looking at tracks with less than 20GeV energy of initial particle, and fitting the above regression, one obtains energy resolution about 0.27. The data and it's fit are presented fig. 13

Acknowledgements

We would like to thank OPERA collaboration and Prof. Giovanni De Lellis, spokesperson of the collaboration, for sharing the data sample and supporting this challenge.

Vlad's solution

First step. Background Filtering. For each track T_0 in layer N find two possible sequential tracks in the layer $N+1$ with the lowest integral distance between them (fig. 14), T_1, T_2. For this combination (T_0, T_1, T_2) a set of features is constructed: integrated distance, $\Delta\theta$ per candidate, *chi*2 of candidates, $\chi 2$ of track. Classifier: BDT (LightGBM). Precision-Recall curve is at (fig. 15).

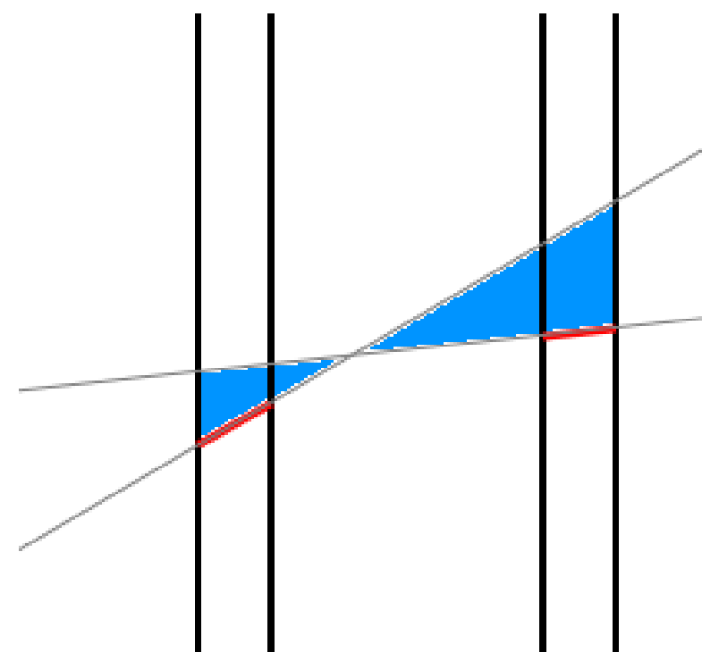

Figure 14: Area of the blue segment is a measure of similarity between two tracks

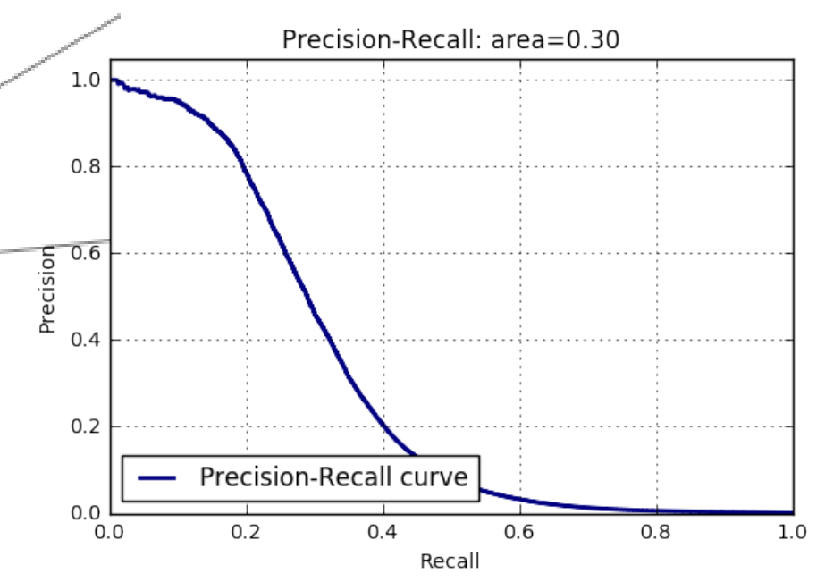

Figure 15: precision/recall for first step in full brick

Second step. Find shower patterns using Conditional Random Field. CRF exploits relations between tracks expressed in a form of energy potentials.

To set up CRF model one needs to:

1. define unary potential: prior knowledge whether the track is signal or background
2. define pairwise potentials: similarity between two tracks

After energy minimization procedure we get the model that discriminates signal from background. The performance of CRF:

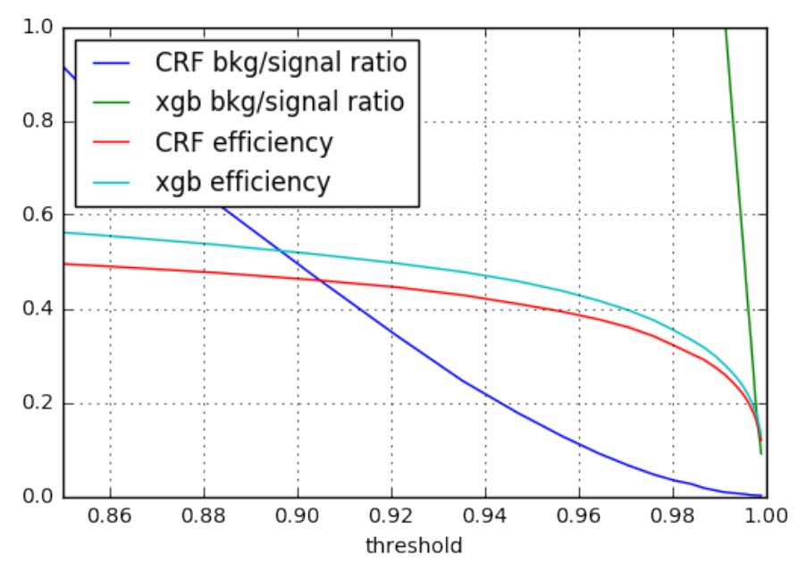

Figure 16: Background / signal ratio and efficiency

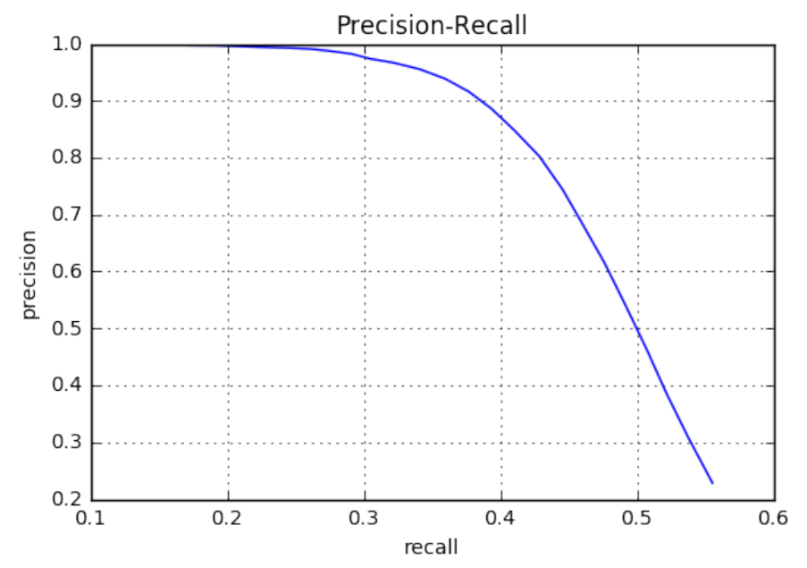

Figure 17: Precision-recall curve in full brick

CRF gives a significant reduction in noise/signal ratio with small decrease in efficiency. Energy resolution ≈ 0.27.

Conclusion

1. All three solutions show the possibility to find the shower without apriori information or the shower origin;
2. comparable quality in terms of Precision/Recall;
3. Solutions presented are capable of finding more than one shower in the brick.

References

[1] https://github.com/catboost

[2] Hosseini B 2014

[3] Muller A C and Behnke S 2014 *J. Mach. Learn. Res.*