Machine Learning for electromagnetic showers reconstruction in emulsion cloud chambers

S.Shirobokov[1,2], **A.Filatov**[1,2], **V.Belavin**[1,2,3], **A.Ustyuzhanin**[1,2]

[1]National Research University Higher School of Economics, Myasnitskaya 20, 101000 Moscow, Russia

[2]Yandex School of Data Analysis, Timura Frunze 11 bld 2, 119034 Moscow, Russia

[3]Moscow Institute of Physics and Technology, Institutskiy per. 9, Dolgoprudny, Moscow Region, 141700, Russia

E-mail: shirobokov@yandex-team.ru, afilatov@hse.ru, belavin@phystech.edu

Abstract. Traces of electromagnetic showers in the neutrino experiments may be considered as signals of dark matter particles. For example, SHiP experiment is going to use emulsion film detectors similar to the ones designed for OPERA experiment from dark matter search. The goal of this research is to develop an algorithm that can identify traces of electromagnetic showers in particle detectors, so it would be possible to analyse and compare various dark matter hypothesis. Both real data and signal simulation samples for this research come from OPERA experiment. Also we have used exploited algorithm for electromagnetic showers identification as a baseline. Although in this research we have used no hints about shower origin.

1. Introduction

Neutrinos are invisible in the detector, but a very small fraction of them will interact with material in the detector and produce an electromagnetic shower in case of a scattered electron and also a hadron shower in case of a scattered nucleus. Atomic electrons can also be scattered by light dark matter which is searched for by other experiments, including those at CERN, for example at SHiP [1] which is going to use very similar neutrino detection techniques as OPERA experiment [2]. The goal of the research is to find traces of electromagnetic showers within a very dense background sample. Baseline algorithm used by OPERA [3] relies on the known origin of the shower, but in our case we are not going to use any a priori knowledge of the shower origin. Several solutions to the problem are suggested.

2. Data sample

The OPERA experiment data samples are structured corresponding to physical bricks of photo emulsion (emulsion cloud chamber). Each brick consists of 57 emulsion films. Each of the film plays the role of calorimeter and is used for track detection from electromagnetic shower, originated from the point of neutrino interaction. Each track[1] is described by 3 coordinates and 2 direction variables (slopes of the track in X/Y projections, assuming that xy-plane is parallel to emulsion films) as well as goodness of fit parameter χ^2. The total background studied in

[1] Mostly we are using word "track" to describe hits in emulsion films, but sometimes we refer to "track" as real particle trace, especially when discussing shower origin and energy estimation.

this work is $\sim 2 \cdot 10^6$ tracks, which constitute one tenth of the background tracks per event comparing to the real background sample. Signal sample representing electromagnetic samples is simulated.

3. Artem's solution

The algorithm proceeds in two steps: at the first step we would like to clean the brick saving as much signal as possible. At the second step, the algorithm classifies remaining track sequences and detects the background which was not detected at the first step.

3.1. SVM step

At the first step of the algorithm we train Support Vector Machine (SVM) [4] to classify pairs of tracks: SVM should indicate whether the pair belongs to signal or background class. We compute the following features: impact parameters to both direction, euclidean distance between tracks, tangent of angle, difference in projection angles to X and Y, χ^2 for both tracks.

The algorithm can be described by the following pseudo-code.

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 the probability of each neighbor to be a continuation of the track using SVM;
        if the highest probability is bigger than threshold or the track has more than h
         ancestors then
            Leave the track;
        else
            Delete the track;
```

The neighbors on the next layer are tracks that lie inside the square of the given area with the center in the continuation of the predecessor track.

3.2. CRF step

After the SVM step we get a cleared brick where only sequence-like structures are left. In order to clean the brick from the background which forms linear structures we run the Conditional Random Field (CRF) [5] algorithm to classify the sequences.

It is important to note that CRF learns the sequences of the edges, not the sequence of tracks. We use this approach, because coordinates of tracks do not contain any information which could help us to classify the sequences, whereas transition features between tracks can help us a lot. From the cleared brick we have created sequences using SVM: if classifier indicates that two tracks are a part of one sequence with the high probability, then we concatenate them to a single sequence.

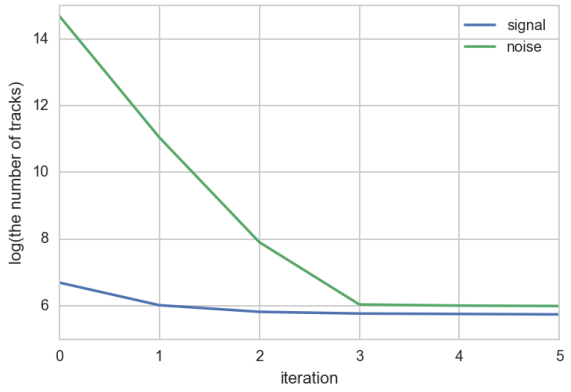

Figure 1. Dynamic of signal and background in the SVM step.

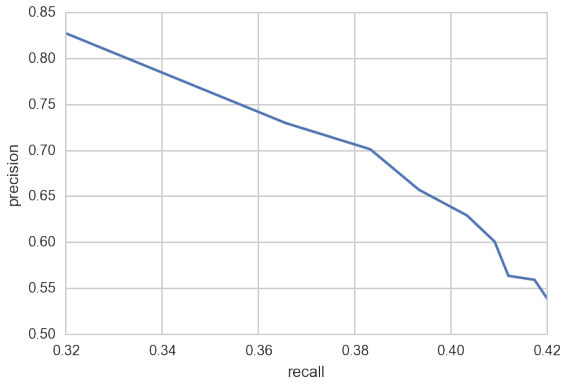

Figure 2. Precision/recall curve for SVM step (Full brick on last iteration).

3.3. Experimental results

In the experiment we applied SVM step multiple times to facilitate the work at the CRF step. On the figure 1, one can see that after multiple SVM steps number of noise decreases dramatically, whereas the signal decreases very slowly. The figure 2 demonstrates the trade-off between precision and recall for the SVM step. Applying CRF step we got the following results. The mean precision/recall for the second step is about 0.48/0.9. Energy resolution is 0.43.

4. Sergey's solution

4.1. Location of shower origin. Initial classification

Since the original dataset is highly imbalanced - highly enriched with background, the main purpose of the first stage is to discard it as much as possible. For that, the new features are created.

Each layer of emulsion is considered with its two neighbor layers. For each track a projection of its direction is obtained for both layers. All the tracks belonging to a small rectangle of size R over the track projections to both layers are selected. Then, the Impact Parameter (IP) projection between given track and selected tracks are calculated, and K tracks with smallest IP are selected. In addition to given features, χ^2 differences and χ^2 itself are added as features. In total there are 81 features for each track. After that, XGBoost classifier (XGB) [6] with weighted penalty function is trained. The weights are shifted towards signal tracks, to penalize classifier more, if it makes mistakes on signal events.

4.2. Topological filter

After the first stage, there is still a large amount of background left. However, the key point here is that in signal events the tracks are clustered near the real shower tracks, whereas background tracks are distributed evenly over the brick volume. Using this fact, 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. A typical plot of median value is shown on fig. 3. As one can see there is a plateau, corresponding to the tracks, that form the real shower. Then the algorithms proceeds to a couple of stages, firstly detecting plates, corresponding to the plateau, and then deleting tracks in each layer if it lies away from the median. After that, only tracks in the center of shower are selected. By calculating series of center of selected tracks for each layer, and fitting Principal Component Analysis (PCA) on them, selecting first component, one gets the direction of the shower.

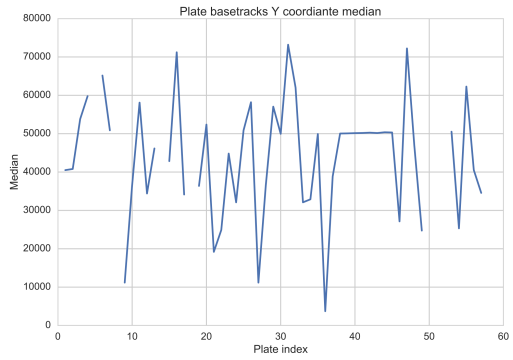

Figure 3. Plate tracks Y median values as a function of plate number.

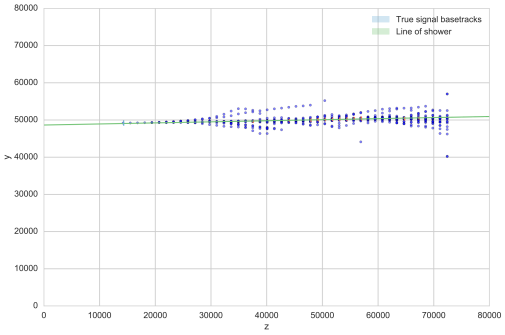

Figure 4. Typical (Y,Z) view of brick after clearance. Red points are selected as signal. Blue is the true signal. Green is shower line.

4.3. Shower origin search

Now, algorithm proceeds iteratively. 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. The algorithm supposes that shower origin is at plate N-2, finds tracks in smaller region near the point at plate N-1 and calculate IP to "supposed" origin. Tracks with IP smaller than $threshold_1 = 150\mu m$ are considered. If there are no such tracks, tracks with bigger IP, but with small angle w.r.t. "origin" are considered. If there are such tracks, plate N-1 assumed to contain signal tracks and plate N-2 marked as non-origin. The process is repeated, starting from plate N-1. The iterations stop once there were no suitable tracks found at plate N-2.

The idea of the search above is that IP feature of the signal tracks near shower origin is distributed highly distinguishable from the IP of the background tracks.

The algorithm has run over 400 events and more than 200 tracks were selected. The mean distance error in XY plane is 0.4 mm, and mean distance error in Z coordinate is 2.4 mm. About 85% of the showers origins are detected within 2 plates distance from the true origin.

5. Vlad's solution

Shower identification is conducted in two separated steps. Firstly, LightGBM [7] classifier (gradient boosting framework that uses tree based learning algorithms) is applied in order to reduce background/signal ratio. Secondly, conditional Random Field (CRF) model is trying to find patterns in the data.

5.1. First step. Background Filtering

First of all for each track T_0 in the layer N we are searching for two possible successive tracks T_1, T_2 in the next layer $N+1$. To measure likelihood that some track T_i in the layer $N+1$ is successive to the track T_0 from the layer N the integral distance between pair of track is calculated (fig. 5), thus two tracks with the lowest integral distance are chosen as successive.

Secondly, for the track T_0 a set of features is constructed: integral distances $d(T_0, T_1)$ and $d(T_0, T_2)$; $|\theta_{X,T_0} - \theta_{X,T_1}| + |\theta_{y,T_0} - \theta_{y,T_1}|, |\theta_{X,T_0} - \theta_{X,T_2}| + |\theta_{y,T_0} - \theta_{y,T_2}|$, where $\theta_{x,T}$ is a slope between of track T in X projection; $\chi^2_{T_0}, \chi^2_{T_1}, \chi^2_{T_2}$.

LightGBM classifier is trained on the generated data. For this model Precision Area Under Curve (PR-AUC) in a brick equals 0.30. By varying threshold, we are changing both efficiency and background/signal ratio, which is crucial at the second step.

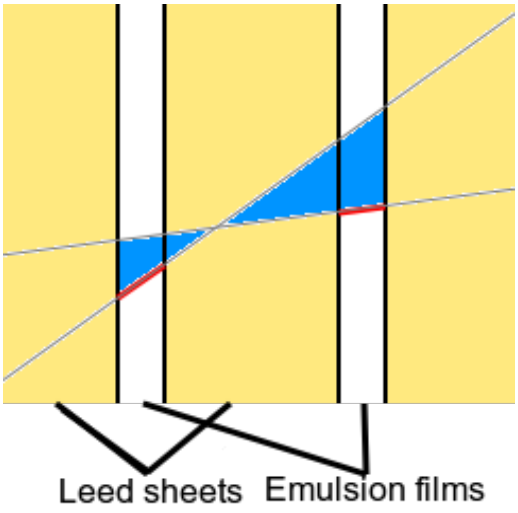

Figure 5. Red segments represent tracks. Area of the blue segment is a measure of similarity between two tracks.

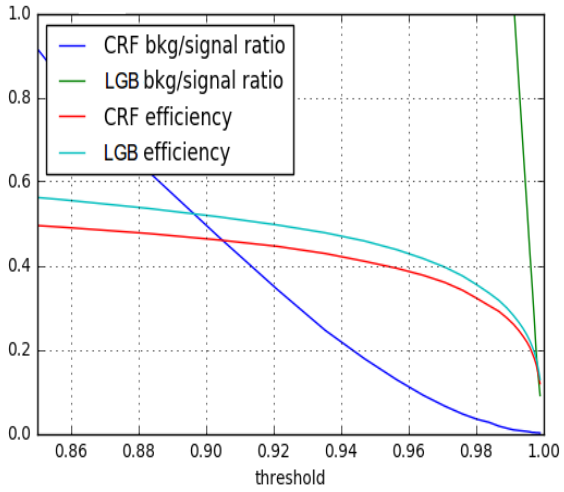

Figure 6. Background to signal ratios and efficiencies as a function of threshold. LGB stands for LightGBM.

5.2. Second step. Find shower patterns using CRF

At the second step, CRF model is applied to the filtered data. The main advantage of the CRF is an ability to take into account relations between tracks.

The underlying idea of the CRF model is an energy minimization of the following potential:

$$E = \sum_i \left(t_i\theta_i(1) + (1-t_i)\theta_i(0)\right) + \sum_{(i,j)\in Edges} \left(t_i(1-t_j)\theta_{ij}(1,0) + t_j(1-t_i)\theta_{ij}(0,1)\right),$$

where t_i is a class label of track T_i; *Edges* is a set of such pairs of indices (i,j) such that if T_i lies in N plate, then T_j should lie in $N+1$ or $N+2$ plate and $d(T_i,T_j) < 5\cdot 10^5$.

To set up CRF model one needs to define unary potentials $\theta_i(1)$ and $\theta_i(0)$ that represent prior knowledge whether the track is a signal or background. For this problem we define them in the following way: $\theta_i(0) = 0$, $\theta_i(1) = -\left(\log p(t_i=1) - C\right) + C\cdot\log(1-N_i)$, where C is a 25th percentile over $\{p(t_i=1)\}_{i=1}^N$, N_i is a number of tracks T_j that lie in one of the two following layers and satisfy constraint $d(T_i,T_j) < 5\cdot 10^5$, and $p(t_i=1)$ is a probability obtained from GBT classifier trained at the first step. Pairwise potentials $\theta_{ij}(1,0), \theta_{ij}(0,1)$ represent similarity between two tracks, thus $\theta_{ij}(0,1) = \theta_{i,j}(1,0) = 5\cdot 10^5/d(T_i,T_j)$.

Graph construction and energy minimization are done by PyMaxflow library [8]. On the (fig. 6) one can see the background reduction effect of CRF model as a function of threshold applied at the background filtering stage. CRF model significantly reduces background/signal ratio with the comparably small decrease in the efficiency.

6. Energy Resolution

6.1. Shower tracks selection

Once the direction and initial point of shower are approximately found, the Hosseni algorithm [3] is applied. Mean average precision (area under curve) is 0.81 ± 0.07.

6.2. Energy resolution
For energy resolution one builds regression of the form

$$E_{rec} = a \cdot N_{st} + b, \quad ER = \sigma\left(\frac{E_{rec}}{E_{true}}\right),$$

where $E_{rec}-$ estimated energy of initial particle, $N_{st}-$ number of selected tracks, $ER-$ is energy resolution, $E_{true}-$ is true energy of initial particle.

Looking at tracks with less than 20 GeV energy of initial particle, and fitting the above regression, one obtains energy resolution about 0.27.

7. Conclusion

Constructed algorithms are tested with one tenth of real background in an event. The algorithms are optimized to work well with showers containing more than 200 tracks. The best energy resolution obtained is at the level of 0.27 that is comparable to the resolution of the baseline algorithm [3], but our algorithms do not rely on the information of the shower origin. With some modifications these algorithms could be scaled for multi-shower setting, which is crucial for SHiP experiment.

8. Acknowledgements

We would like to thank OPERA collaboration and Prof. Giovanni De Lellis, spokesperson of the collaboration, for sharing the data and simulated signal samples.

References

[1] Alekhin S et al 2016 A facility to Search for Hidden Particles at the CERN SPS: the SHiP physics case *Reports on Progress in Physics* 79(12)
[2] Acquafredda R et al 2009 The OPERA experiment in the CERN to Gran Sasso neutrino beam *Journal of Instrumentation* 4.04 P04018
[3] Hosseini B 2015 Search for Tau Neutrinos in tau to e Decay Channel in the OPERA Experiment *Napoli University*
[4] Cortes C Vapnik V 1995 Support vector machine *Machine learning* 20(3) pp 273-297
[5] Lafferty J McCallum A Pereira F 2001 Conditional random fields: Probabilistic models for segmenting and labeling sequence data
[6] Chen T and Guestrin C 2016 Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (ACM) pp 785-794
[7] Ke, Guolin and Meng, Qi and Finley, Thomas and Wang, Taifeng and Chen, Wei and Ma, Weidong and Ye, Qiwei and Liu, Tie-Yan 2017 LightGBM: A Highly Efficient Gradient Boosting Decision Tree *Advances in Neural Information Processing Systems* 30 pp 3149-3157
[8] Boykov Y and Kolmogorov V 2004 An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision *In IEEE Transactions on PAMI* 26(9) pp 1124-1137