Machine learning predictor for 'measurement-to-track' association for the ATLAS Inner Detector trigger

Nisha Lad on behalf of the ATLAS Collaboration

University College London, Gower St London, WC1E 6BT, United Kingdom

E-mail: nisha.lad.13@ucl.ac.uk

Abstract. The track finding algorithm adopted for the LHC Run-2 data taking period is based on combinatorial track following, where the number of seeds scales non-linearly with the number of hits. The corresponding CPU time increase, close to cubical, creates huge and ever-increasing demand for computing power. This is particularly problematic for the silicon tracking detectors, where the hit occupancy is the largest. Therefore, it is essential that resource use is reduced, whilst maintaining the ability to reconstruct tracks with minimal loss in efficiency. This paper briefly summarises the work that has been done to optimise the HLT ID track seeding software for ATLAS Run-3 and beyond, in order to reduce the number of fake seeds constructed. An ML-based algorithm has been developed to predict if a pair of hits belong to the same track given input hit features, focusing on cluster width and inverse track inclination. The implementation of the trained predictor in the form of Look-Up Tables is presented, along with performance results in terms of tracking efficiency and speed-up factor using simulated data.

1. Introduction

ATLAS [1] is a general-purpose detector at the Large Hadron Collider (LHC). The Inner Detector (ID) trigger is part of the ATLAS High Level Trigger (HLT) system and performs fast online track and vertex finding. It consists of the `Fast Tracking` stage, providing initial track reconstruction using seeded track finding and combinatorial track following [2]. Typically, a seed is defined as a group of three spacepoints located in different detector layers, where a fake seed is defined as a group of spacepoints that do not originate from the same simulated track. From a geometrical perspective, track finding is a pattern recognition problem with the aim of associating individual measurements into sequences. The training is focused on pixel-detector layers, being closest to the beamline and thereby an advantageous region to reduce the proportion of fakes. The classifier predictions are used as input in the `Fast Tracking` algorithm.

2. Measurement-to-track Association

2.1. Data Exploration & Feature Extraction

Seeds constructed at the combinatorial stage of ATLAS track seeding from the Run-2 geometry were used to extract hit-pairs to form a training dataset. Monte Carlo (MC) $t\bar{t}$ samples with centre-of-mass energy $\sqrt{s} = 13$ TeV at mean pile-up interaction multiplicity $<\mu> = 80$ were used. An illustration of a seed in the ID pixel layers is shown in Figure 1. From each triplet seed, the inner doublet (hits 1, 2) and outer doublet (hits 2, 3) are extracted. For each doublet the minimum and maximum absolute inverse slope of the track, $|cot(\theta)|$, were calculated using r-z coordinates and used as an input feature for training. θ is the angle of inclination of a hit-pair with respect to the z axis. The longitudinal pixel cluster width, w_η, measured in the η direction was also extracted. η is the pseudorapidity defined as $\eta \equiv -ln[tan(\theta/2)]$. The MC generated data in the $[|cot(\theta)|, w_\eta]$ phase space behaves as a set of 1-dimensional distributions, each with discrete w_η. This characteristic is exploited to form an ensemble of predictors.

MC truth for seeds and their corresponding doublets were extracted from ATLAS track seeding and used as targets in training. Doublets with correct hit association were defined as truth 1, for which its hit-pairs belong to the same track. Conversely doublets with incorrect hit association defined as truth 0, for which its hit-pairs do not belong to the same track. Hit-pairs for the pixel barrel and endcap are handled separately in order to build regional classifiers. A similar methodology was used for both, this paper outlines the approach taken for the barrel.

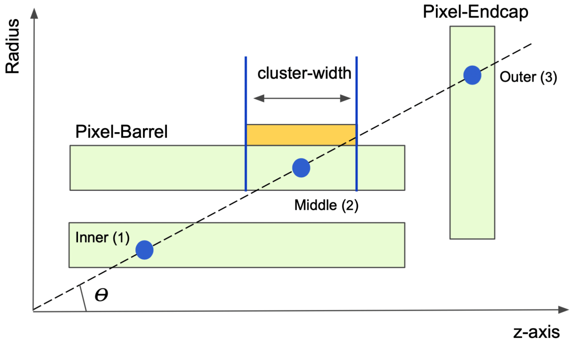


Figure 1: Seed illustration in r-z plane (mm) of the ID. The inner doublet consists of hits 1, 2 and the outer doublet consists of hits 2, 3. The longitudinal pixel-cluster width w_η (mm) is measured in the direction of η, where θ is the angle of inclination with respect to the z axis.

2.2. Classifier Development

The data was split into a training and a test dataset using a 70:30 split. The training data was plotted in the phase space of $[|cot(\theta)|, w_\eta]$ where discrete 1-dimensional bands were observed; this is a direct result from the simulation of the pixel geometry on the modules. Figure 2 shows the 1-dimensional distributions of $|cot(\theta)|$ for $w_\eta \leq 0.4$ mm; a clear distinction can be seen between the two classes. In order to discriminate between the correct and incorrect hit association, Bayes' theorem was used, as it relies on having a prior probability of belief and a conditional likelihood. The Bayes' classifier was implemented in a generative way, as the likelihood was computed via a Kernel Density Estimate (KDE) fitted to each correct hit association distribution of discrete w_η. The choice of bandwidth for the Gaussian kernel was optimised and determined using Silverman's method in order to prevent over- or under-fitting [3]. Therefore, each model was trained as follows; the feature vector $\mathbf{x}$ comprising $|cot(\theta)|$ and the truth label of the doublet formed the target vector $\mathbf{y}$. For unknown data points, the class which maximised the posterior was the class prediction.

After training, each classifier's predictions were tuned using its corresponding Receiver Operating Characteristic ROC curve [4] to yield a True Positive Rate (TPR) of 0.95 in order to maintain a high purity of correct hit-pairs. Figure 4 shows the adjusted classifier predictions for each distribution for pixel-barrel hits. A 2-dimensional acceptance-rejection region is constructed; the black region of acceptance shows doublets predicted with correct hit association, which follow a linear corridor trend with a moving average as the cluster width increases. Regions of high rejection are predicted as areas with low pixel cluster width and large track inclination, as well as large pixel cluster width and small track inclination. This is to be expected to originate from the forward region of the detector. Overall, the recall (and TPR) achieved for doublets with correct hit association was determined to be a tuned 95%. The purity achieved was 56% and the F1 score 71%. For each classifier, the triplet selection efficiency and total rejection rate were also evaluated in order to give an indication of speed-up factor.

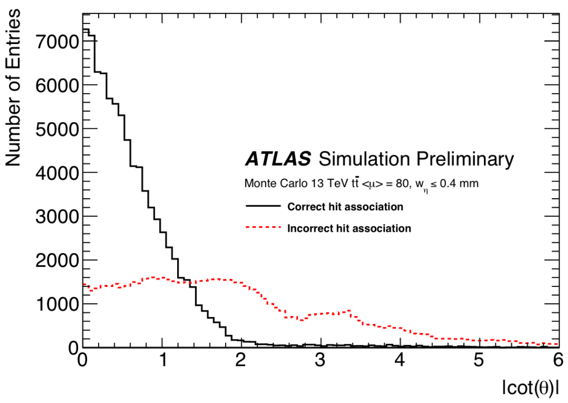


Figure 2: $|cot(\theta)|$ distribution for truth pixel-barrel doublets from MC 13 TeV $t\bar{t} <\mu>= 80$ samples with $w_\eta \leq 0.4$ mm. w_η the pixel cluster width measured in the η direction, θ is the inclination angle of the doublet with respect to z [5].

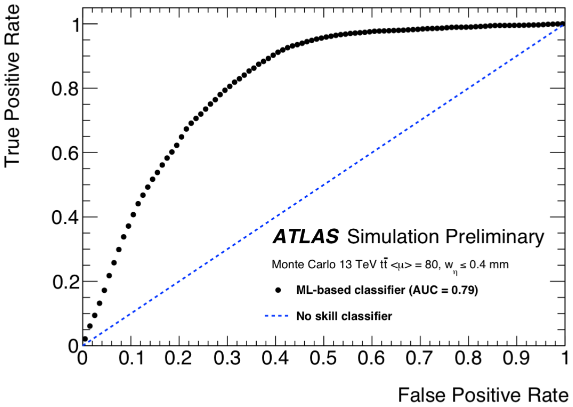


Figure 3: ROC curve generated from KDE-based classifier predictions on pixel-barrel doublet data for $w_\eta \leq 0.4$ mm. Area Under Curve (AUC) achieved is 0.79. The model is fitted with Gaussian KDE, trained on MC 13 TeV $t\bar{t} <\mu>= 80$ samples [5].

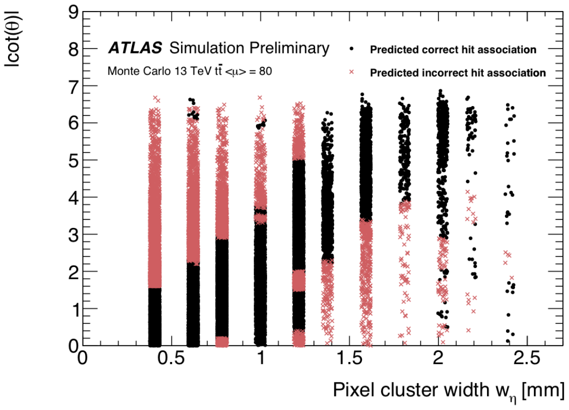
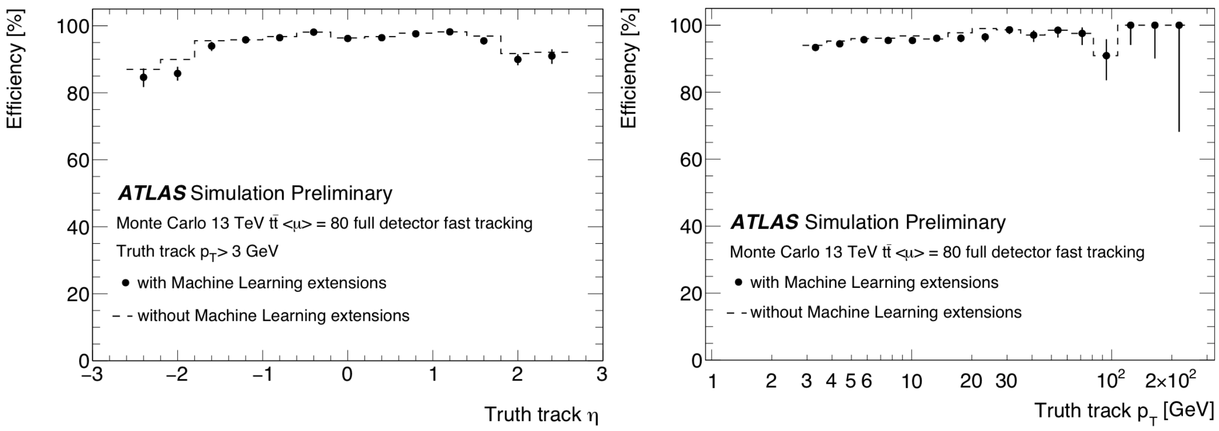


Figure 4: Predicted classification of pixel-barrel hit-pairs from the ATLAS pixel detector to tracks corresponding to truth particles. $|cot(\theta)|$, where θ is the inclination angle of the doublet with respect to z, vs pixel cluster width measured in the η direction, w_η. Classifiers were trained using pixel-barrel doublets MC 13 TeV $t\bar{t} < \mu >= 80$ samples and used to discriminate between doublets which have correct and incorrect hit association [5].

3. Performance Evaluation

Classifier predictions for the pixel barrel (Figure 4) and endcap regions were implemented directly into the ATLAS HLT `Fast Tracking` to reduce computational overheads. This was achieved by converting the predefined acceptance region into a Look-Up Table (LUT) using a common image processing technique known as Morphological Filtering [6].

A comparison of full scan ID efficiencies as a function of MC truth track η and p_T, for the nominal seeding approach and with application of ML filtering within the pixel detector, are shown in Figures 5a and 5b. The nominal seeding approach achieved an average tracking efficiency of 95% with respect to MC truth tracks. With application of ML filtering at $< \mu > =$ 80, an average tracking efficiency of 93.9% was observed and speed-up by a factor of 2.3. The main loss in efficiency is observed at regions of large $|\eta|$, which may be a direct result of material effects from the forward detector region.



(a) Efficiency vs. MC truth track η.

(b) Efficiency vs. MC truth track p_{T}.

Figure 5: Tracking efficiencies as a function of track parameters for $p_{\mathrm{T}} > 3$ GeV, for the ATLAS full detector tracking with $t\bar{t} < \mu > = 80$. The data points show the efficiency when using ML extensions in the seed building stages of `Fast Tracking` in the ATLAS pixel detector, prior to track fitting. The errors shown are purely statistical [5].

3.1. CPU Time Comparison

Table 1 summarises the breakdown in speed-up factor achieved at various stages within the `Fast Tracking` trigger algorithm, with the application of the trained LUT for the pixel region. The greatest saving in CPU time is achieved during the `Seed Processing` stage, as a direct result of a significant reduction in the number of seeds. The average number of seeds processed for a given region of interest for the standard tracking is $O(10^4)$, whereas with the introduction of ML filtering for pixel seeds, about 78% fewer seeds were observed. This significant reduction in CPU time does not only benefit the `Seed Processing` stage of the combinatoric track following, but also propagates to `Track Fitting`.

Table 1: Performance of the ATLAS full detector tracking with MC 13 TeV $t\bar{t}$ samples at $< \mu >= 80$, with the application of ML extensions for filtering on pixel detector hit-pairs in the `Fast Tracking` trigger stage [5]. The total speed-up factor and breakdown of speed-ups at different stages of the `Fast Tracking` trigger algorithm are presented, each speed-up is presented with respect to the standard trigger seeding where no ML extensions were applied.

Total Speed-up Factor	Seed Generation	Seed Processing	Track Fitting
2.3	1.3	3.3	1.5

3.2. Changing Pileup Conditions

Table 2 summarises the relative efficiency loss and relative speed-up factor, with application of ML extensions for the ATLAS pixel detector in various mean pile-up interaction multiplicities $< \mu >$. The speed-up factor increases as $< \mu >$ increases with minimal loss in efficiency.

Table 2: Performance of the ATLAS full detector tracking with MC 13 TeV $t\bar{t}$ samples at $< \mu > = 40$, 60 and 80, with the application of ML extensions for filtering on pixel detector hit-pairs in the `Fast Tracking` trigger stage prior to the track fitting [5]. The absolute loss in average tracking efficiency and the total speed-up factor for seeded track finding in the ATLAS pixel detector are presented with respect to the standard trigger seeding where no ML extensions were applied. The efficiency loss is mainly observed at large $|\eta|$. The statistical uncertainties in efficiencies are $O(10^{-3})$, hence are not quoted.

$< \mu >$	Efficiency Loss (%)	Total Speed-up Factor
40	0.7	1.6
60	0.7	2.1
80	1.1	2.3

4. Summary

The application of a ML-based classifier for seed selection in the ATLAS ID has provided significant CPU savings on trained MC data at various pileup levels. The trained predictor in the form of a LUT yields 2.3× speed-up with minimal loss in efficiency (1.1%) at $< \mu >= 80$ compared with the standard trigger tracking. The developed ML pipeline provides a way to generate custom LUTs by training the predictor to yield a required TPR, dependent on the degree of efficiency required. Reducing the proportion of fakes at an earlier stage in the ATLAS HLT track seeding, ensures the reduction in CPU usage overall. Efficient use of computing power will become an increasingly important factor in the selection of physics objects as the luminosity and pileup increase during future upgrades of the LHC program.

References

[1] The ATLAS Collaboration. "The ATLAS experiment at the CERN large hadron collider". In: *J. Instrum.* 3.08 (2008), S08003–S08003.

[2] Carlo Schiavi on behalf of the ATLAS Collaboration. *ATLAS High-Level Trigger algorithms for Run-2 data-taking.* `https://cds.cern.ch/record/2016651/files/ATL-DAQ-PROC-2015-018.pdf`. Accessed: 2022-1-11. 2015.

[3] M C Jones, J S Marron, and S J Sheather. "A brief survey of bandwidth selection for density estimation". In: *J. Am. Stat. Assoc.* 91.433 (1996), p. 401.

[4] Jesse Davis and Mark Goadrich. "The relationship between Precision-Recall and ROC curves". In: *Proceedings of the 23rd international conference on Machine learning - ICML '06.* New York, New York, USA: ACM Press, 2006.

[5] *HLT Tracking Public Results Atlas Public TWiki.* en. `https://twiki.cern.ch/twiki/bin/view/AtlasPublic/HLTTrackingPublicResults`. Accessed: 2022-1-11.

[6] *Morphological Filtering — skimage v0.12.2 docs.* en. `https://scikit-image.org/docs/0.12.x/auto_examples/xx_applications/plot_morphology.html`. Accessed: 2022-1-11.