

LHCb Topological Trigger: Approaches and Optimization

Tatiana Likhomanenko^{1,2,3} tatiana.likhomanenko@cern.ch, Philip Ilten⁵,
Egor Khairullin^{1,4}, Alex Rogozhnikov^{1,2}, Andrey Ustyuzhanin^{1,2,3,4}, and
Michael Williams⁵ mwill@mit.edu

¹ Yandex School of Data Analysis (YSDA), RU

² National Research University Higher School of Economics (HSE), RU

³ NRC "Kurchatov Institute", RU

⁴ Moscow Institute of Physics and Technology, Moscow, RU

⁵ Massachusetts Institute of Technology, US

Abstract. The LHCb trigger system plays a key role in selecting events (proton-proton collisions) to store them into memory for further processing. It should choose several thousands events per second among all detected by LHCb. The main b-physics trigger algorithm (should select events with B-meson decays) used by the LHCb experiment is a so-called topological trigger. In the LHC Run 1, this trigger, which utilized a simple boosted decision tree algorithm, selected a nearly 100% pure sample of b-hadrons with a typical efficiency of 60-70%; its output was used in about 60% of LHCb papers. In the paper [1] we presented studies carried out to optimize the topological trigger for LHC Run 2. Trigger data have a specific structure and that is why necessary quality measure was proposed. In this paper we investigate the presense of noisy samples in data and propose several ways to reduce this noise. Another side of the trigger system is a real-time prediction operation: it should be very fast. In the paper [1] two approaches how to speedup a prediction operation and to preserve the quality as possible were considered: bonsai boosted decision tree format (used in Run 1) and decision trees post pruning. In this paper we analyse behaviour of the cleaned up samples for these two approaches. As a result, we demonstrate that removing of noise can still improve reoptimized topological trigger on the Run 1 performance for a wide range of b-hadron decays.

Keywords: high energy physics, machine learning, LHCb trigger system

1 Introduction

The LHCb detector [2] is designed for studying beauty and charm (heavy flavour) hadrons produced in proton-proton collisions at the Large Hadron Collider (LHC) at CERN. At LHCb data were collected with the rate 40 MHz in Run 1 ($40 \cdot 10^6$ proton-proton collisions, called events, per second). This amount cannot be store into memory. A trigger system hardware part, called Level-0 (L0), reduced the

visible bunch crossing rate to 1 MHz at which the detector could be read out. Further, a flexible software High-Level Trigger (HLT) applied a range of more advanced selections that reduce the rate to about 5 kHz ($5 * 10^3$ events per second) for offline storage and processing. This configuration allowed LHCb to record the largest beauty and charm hadron samples at a very high signal purity. The performance of the Run 1 trigger is described in detail in [3], [4].

In Run 1 it was shown that complex, multivariate trigger selections were possible. The HLT processes few enough events that it is possible to perform the reconstruction of a collision (to reconstruct tracks, vertices, a topology, physical characteristics). There are many HLT lines dedicated to triggering on various types of events. The majority of analyses using b-hadrons at LHCb made use of the topological n-body trigger [5]. Most n-body hadronic B decays ($n \geq 3$) are only triggered on efficiently in LHCb by these lines. The topological n-body trigger is an inclusive trigger which combines successive 2,3 and 4-body track combinations. A novel boosted decision trees (BDT) multivariate selection [6] was used in Run 1.

In the [1] we studied upgrading of the topological trigger: a new scheme of the trigger processing, which includes a more widespread usage of machine learning, was proposed; the appropriate quality measure was considered (receiver operating characteristic, ROC, computed for events for all B decays). However, obtained algorithms cannot be used in real-time trigger data processing. That is why, in the paper [1] we compared two approaches how to speedup a prediction operation: bonsai boosted decision tree format [6] and decision trees post pruning.

In the previous analysis [1] simulated data for various B decays were used as signal-like samples and real data from the collider without any physics were used as background-like samples. In this paper we present studies connected with the fact that simulated samples contain noisy data. We propose several ways how to clean up the samples and analyse the quality of the two speedup approaches in this case. As a result, we demonstrate that removing of noise can still improve the reoptimized topological trigger on the Run 1 performance for a wide range of b-hadron decays.

2 Motivation

Data for the trigger system have a specific structure: each event consists of the several reconstructed secondary vertices; a set of all secondary vertices for all B decays are the training sample. After training an event will pass the trigger if at least one its secondary vertex will pass the trigger (passing the trigger means that a prediction for a sample is greater than some fixed threshold; a threshold is chosen in a way that the false positive rate value provides the trigger rate). And we are interested in the greatest true positive rate for each B-decay mode.

As background-like data we use real data from the collider without any physics (that is why choosing the trigger rate will be equivalent to choosing the false positive rate). It means that all reconstructed secondary vertices for each event

in the background-like data don't contain any interesting decays. As signal-like data we use the proton-proton collisions simulated for each mode of B-decays. For these data we really know that at least in one secondary vertex necessary decay happened. But it doesn't guarantee that all secondary vertices contain interesting B-decays.

Each event can contain up to 40 secondary vertices. For the signal-like data one of these is a truly signal-like sample, but others can be noise.

3 Noise clean up approaches

4 Speedup approaches with cleaned up samples

5 Conclusion

In the HLT1 during Run 1, the so-called "1 track" line was used because there was not sufficient CPU time for combinatorics and because the tracking transverse momentum threshold was too high for an secondary vertex (SV) based approach to be efficient. In Run 2, the HLT1 tracking transverse momentum threshold will be the same as was used in the HLT2 for Run 1. This makes it possible to use, along with a "1 track" line, a "2 body SV" line. Thus, we consider a new HLT scheme (figure 1). The combination of displacement from the primary vertex (PV) and high transverse momentum are the main characteristics of interesting decays. The HLT "1-track" line is looking for either one super high transverse momentum or high displacement track. The HLT "2 body SV" line uses a multivariate analysis to look for two tracks that form a vertex. The HLT topological line is improved using a more powerful multivariate analysis. It uses fully reconstructed events to look for 2, 3, and 4 tracks making a vertex.

The analysis was conducted on the following data: signal samples are simulated 13-TeV B-decays of various topologies, while the background sample is generic Pythia 13-TeV proton-proton collisions.

Fig. 1. HLT Run 2 scheme. HLT1 consists of two lines: the "1-track" and "2 body SV", with corresponding output rates 100 kHz and 50 kHz. If an event passes at least one of these two lines, then it is triggered by the topological line with output rate 2-4 kHz.

[width=16pc]images/scheme.pdf

6 Noise clean up approaches

Each event is represented as set of secondary vertices. These secondary vertices are the input data to the classifier. The event is triggered if at least one secondary vertex passes the classifier selection. A trigger's output rate is limited.

This restriction is equivalent to a restriction on the background events efficiency (FPR). For example, for an output rate of 2 kHz, then FPR is 0.2%. This can be shown using ROC curves from the classifier, which are plotted for test events (figure 2).

All classifiers were trained on one half of the data and were tested on the other half. For the HLT "1-track" and "2 body SV" lines, all signal modes were used, while for the HLT topological line six specific modes were used in training. All algorithms were tested on all available signal modes.

6.1 HLT "1-track" line

Preselections and variables for the HLT "1-track" line are listed in table 1. Only two variables are used in this line. To optimize the signal efficiency and find an optimal decision boundary, multivariate analysis is used. Distributions of signal and background are shown in figure 3. The MatrixNet[7] algorithm (figure 5), logistic regression (figure 4) and neural networks (figure 6) were trained to find the decision boundaries. From experiments, MatrixNet decision boundary is found to be the best, but for online processing a more simple decision rule fits: a hyperbolic function can be used as the decision boundary. Efficiencies comparison of different algorithms for B-modes is shown in figure 7.

Fig. 2. Trigger events ROC curve. An output rate of 2.5 kHz corresponds to an FPR of 0.25%, 4 kHz — 0.4%. Thus to find the signal efficiency for a 2.5 kHz output rate, we take 0.25% background efficiency and find the point on the ROC curve that corresponds to this FPR.

[width=22pc]images/roc_events.pdf

[width=18pc]images/track-db

Fig. 3. Track data scatters, described in two-dimensional space.

[width=18pc]images/mn-track-db

Fig. 5. Decision boundary for MatrixNet algorithm.

[width=18pc]images/log-track-db

Fig. 4. Decision boundary for logistic regression.

[width=18pc]images/nn-track-db

Fig. 6. Decision boundary for neural networks.

6.2 HLT "2 body SV" line

Preselections and variables for the HLT "2 body SV" line are listed in table 2. The line looks for two tracks that form a vertex. In this case, the MatrixNet algorithm is used and several studies were conducted. Firstly, the possibility to remove the corrected mass cut ($m_{cor} < 10$) and to remove the corrected mass as variable from classifier's input was investigated. The impact on the performance is negligible. These removals were done to reduce systematic uncertainties and to help in exotic searches. Secondly, to minimize the set of variables used in trigger, a selection was conducted among the scalar sum, vector sum, and minimum of the transverse momenta. The results obtained (efficiencies comparison) are shown in figure 8.

Table 1. HLT "1-track" line description.

Track preselections:	$PT > 500 \text{ MeV}$
	$IP_{\chi^2} > 4$
	$track_{\chi^2}/ndof < 3$
Analysis variables:	PT, IP_{χ^2}
Output rate:	100 kHz

[width=18pc]images/track.pdf

Fig. 7. Efficiencies comparison for MatrixNet (MN), neural networks (NN) and logistic regression. 50 kHz and 100 kHz output rates are considered.

[width=18pc]images/hlt1-2body.pdf

Fig. 8. Efficiencies comparison for MatrixNet: with corrected mass, without it (MN), with the scalar sum of transverse momentum (sum pt , without min and vector pt). These models were 'converted' to BBDT format and compared. The output rate is set to 50 kHz.

6.3 HLT topological line

The HLT2 topological lines are designed to trigger efficiently on any B decay with at least 2 charged children. It is designed to handle the possible omission of child particles. To save CPU time in the HLT2 reconstruction, only tracks with transverse momentum $PT > 200 \text{ MeV}$ are reconstructed. To reduce the background rate due to ghosts, all tracks are required to have a $track_{\chi^2}/ndof$ value less than 2.5. To reduce the background rate due to prompt particles, all tracks are required to have an impact parameter χ^2 value greater than 4. Due to the inclusive nature of the HLT2 topological lines, this does not mean that all of the B daughters need to satisfy these criteria. The trigger is designed to allow

for the omission of one or more daughters when forming the trigger candidate. The previous topological line for Run 1 is described in [?].

In the HLT topological line used in Run 1, a simple boosted decision tree was used [?] to define interesting secondary vertices. For Run 2, the algorithm is reoptimized. Current preselections and training variables are listed in table 3. The output rate is one of the signal efficiencies factor. The efficiency dependence on the output rate is shown in figure 9. Thus, several modes, including two training ones, significantly depend on the output rate.

Table 2. HLT "2 body SV" line description. **Table 3.** HLT topological line description.

Track preselections:	Track preselections:
$PT > 500 \text{ MeV}$	$PT > 200 \text{ MeV}$
$IP_{\chi^2} > 4$	$IP_{\chi^2} > 4$
$track_{\chi^2}/ndof < 2.5$	$track_{\chi^2}/ndof < 2.5$
SV preselections:	SV preselections:
$PT > 2 \text{ GeV}$	$vertex_{\chi^2} < 10$
$vertex_{\chi^2} < 10$	$1 < mcor < 10 \text{ GeV}$
$1 < mcor \text{ GeV}$	$2 < \eta < 5 \text{ (PV to SV)}$
$2 < \eta < 5 \text{ (PV to SV)}$	$N(\text{tracks with } IP_{\chi^2} < 16) < 2$
Analysis variables:	Analysis variables:
sum PT , $vertex_{\chi^2}$, FD_{χ^2} ,	n, mcor, sum PT , $vertex_{\chi^2}$,
$N(\text{tracks with } IP_{\chi^2} < 16)$	η , FD_{χ^2} , min PT ,
	IP_{χ^2} , $N(\text{tracks})$,
	$N(\text{tracks with } IP_{\chi^2} < 16)$
Output rate: 50 kHz	Output rate: 2-4 kHz

Different boosted decision trees were trained (figure 10). Also some hierarchical algorithm was conducted, a so-called 'blend'. Training data was divided into two parts. On the first part three MatrixNet classifiers were trained, which use only 2, 3 or 4 body decays as inputs. Next, the second part was predicted by the corresponding n-body classifiers. These predictions are considered as new additional input variables. The resulting MatrixNet was trained on this second part using basic variables and these additional ones. This hierarchical algorithm gives an improvement for several training modes.

[width=18pc]images/rates.pdf

Fig. 9. Signal efficiencies for training modes (left) and other available modes (right) for different output rates.

Based on these studies, the MatrixNet (MN) model was chosen. Its efficiency was compared with the Run 1 algorithm. For these six training modes, we obtain

significant relative improvement: 15-60% for 2.5 kHz output rate and 50-80% for 4 kHz (table 4).

[width=18pc]images/blend.pdf

Fig. 10. Comparison of different algorithms: MatrixNet (MN), MatrixNet with SV weights (MN reweight), scikit-learn AdaBoost implementation (AdaBoost), TMVA AdaBoost implementation (TMVA) and some hierarchical algorithm (blend).

7 Online processing

Boosted decision trees are not appropriate for online processing events in triggers due to their low speed. Two approaches exist to overcome this restriction. One is the so-called bonsai boosted decision format[6] (shortly BBDT). Since it is a type of BDT, MatrixNet can be converted into this format. The second approach is post-pruning; the basic MatrixNet classifier includes several thousand trees, and the post-pruning procedure reduces this amount to a few hundred. This results in significant speedup of the prediction rate. Both methods reduce MatrixNet signal efficiencies. In case of the BBDT, we are also limited in the size of BBDT lookup table in RAM. Different BBDT versions for MatrixNet were tried and compared to post-pruning to find the optimal solution for online processing (see figure ??).

Table 4. Ratio of Run-2 over Run-1 for HLT2/HLT1 efficiencies. Note that the denominator is reconstructible with $PT(B) > 2$ GeV, $\tau(B) > 0.2$ ps.

mode	2.5 kHz	4. kHz
$B^0 \rightarrow K^*[K^+\pi^-]\mu^+\mu^-$	1.64	1.72
$B^+ \rightarrow \pi^+K^-K^+$	1.59	1.65
$B_s^0 \rightarrow D_s^-[K^+K^-\pi^-]\mu^+\nu_\mu$	1.14	1.47
$B_s^0 \rightarrow \psi(1S)[\mu^+\mu^-]K^+K^-\pi^+\pi^-$	1.62	1.71
$B_s^0 \rightarrow D_s^-[K^+K^-\pi^-]\pi^+$	1.46	1.52
$B^0 \rightarrow D^+[K^-\pi^+\pi^+]D^-[K^+\pi^-\pi^-]$	1.40	1.86

Interestingly, the preference between the BBDT and post-pruning depends on the chosen output rate. It is clear in figure ??, that the ROCs order depends on the background efficiency (or the output rate). Another study for Run 2 triggers is connected to a timing comparison of BBDT and post-pruning, which is in progress at this moment.

8 Conclusion

The LHCb topological trigger was successfully reoptimized for Run 2: 15-60% efficiency improvement was obtained for 2.5 kHz output rate and 50-80% for 4 kHz. The presented HLT scheme will be applied in Run 2.

Notes and Comments. The part of the topological trigger analysis connected with the machine learning studies and presented in this paper and [1] are open source and can be found on the github (<https://github.com/tata-antares/LHCb-topo-trigger>).

References

1. Likhomanenko T., Ilten P., Khairullin E., Rogozhnikov A., Ustyuzhanin A., Williams M.: LHCb Topological Trigger Reoptimization. *Journal of Physics: Conference Series* 664 (2015) 082025 [arXiv:1510.00572]
2. A. A. Alves Jr. et al.: The LHCb Detector at the LHC, *JINST* 3, S08005 (2008)
3. Aaij, R., *et al.* [LHCb Trigger Group]: The LHCb trigger and its performance. *JINST* 8 P04022 (2013) [arXiv:1211.3055]
4. Aaij R. et al.: Performance of the LHCb High Level Trigger in 2012, *J. Phys., Conf. Ser.* 513, 012001 (2014).
5. Gligorov V., Thomas C., Williams M.: The HLT inclusive B triggers. Technical Report LHCb-PUB-2011-016. CERN-LHCb-PUB-2011-016. LHCb-INT-2011-030, CERN, Geneva, Sep 2011. LHCb-INT-2011-030.
6. Gligorov V., Williams, M.: Efficient, reliable and fast high-level triggering using a bonsai boosted decision tree. *JINST* 8, P02013 (2013) [arXiv:1210.6861]
7. Gulin, A., Kuralenok, I., Pavlov, D.: Winning the transfer learning track of Yahoo's Learning to Rank Challenge with YetiRank. *JMLR: Workshop and Conference Proceedings* 14 (2011) 63 (Yandex MatrixNet).