Particle decay classifiers for the LHCb Run 3 first-level trigger

Presented by: Sean Condon - MIT
Mentors: Michael Williams, Thomas Boettcher, Daniel Craik - MIT

28 September 2020
Simulated LHCb Data allows Training of Classifiers

-The task: update the one-track and two-track high level triggers for run 3 of LHCb, implement optimal triggers into Allen pipeline

-Training data: 180,000 simulated LHCb events (~2 million decays), made up of 6 decay types and one 2018MinBias file for trigger rate calculation

-Cut out signals with low momentum and lifetime: parent particle $\text{PT} > 2 \text{ GeV}$, $\tau > 0.2 \text{ ps}$, $2 < \eta < 5$ (removes ~92% of data)
Training Data Cont.

-Classifier inputs are taken from *LHCb Trigger Reoptimization paper†*

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Analysis Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Track Model</td>
<td>PT, IP_{χ^2}</td>
</tr>
<tr>
<td>2-Track Model</td>
<td>sum PT, vertex ( \chi^2 ), FD ( \chi^2 ), N (# tracks with IP_{χ^2} &lt; 16)</td>
</tr>
</tbody>
</table>

**One-Track Data**

*Testing Data Parameter Distribution*

- True Noise
- True Signal
- Data Displayed = 95.8%

**Two-Track Data**

*Testing Data Parameter Distribution*

- True Noise
- True Signal
- Data Displayed = 94.7%
Catboost Enables Easy Training of BDT Models

- Catboost is a python library for Boosted Decision Tree classifiers, enabling:
  - A variety of training options for BDTs
  - Saving classifiers in JSON format easily read into Allen pipeline
- Total simulated LHCb data is split into 75% training, 25% testing
- Training data 92% noise, so noise decays are weighted to balance training
- BDT model hyperparameters (learning rate, model depth, model iterations) optimized by exploring parameter space.
Notes on Classifier Evaluation

-Catboost Classifiers are trained on single decay examples, but they are evaluated on LHCb events (average ~10 decays) for efficiency and trigger rate.

-For each decay, output = softmax-ed vector in the format of $[\text{Prob}_{\text{noise}}, \text{Prob}_{\text{signal}}]$.

-If any decay in an evaluation event has $\text{Prob}_{\text{signal}} >$ threshold, the entire event is considered to be triggered on.

-True positives when event has signal, AND classifier triggers on signal decay.
Performance of Preliminary Classifiers

**One-Track**: $\text{eff} = 13.7\%$ at 1 MHZ

**Two-Track**: $\text{eff} = 40.7\%$ at 1 MHZ
Addressing Decay Type Imbalance

- Training datasets are very imbalanced by decay type \( N_{\text{large}} = 95,000 \) (Bs2JPsiPhi); \( N_{\text{small}} = 2,470 \) (Ds2KKPi)

- Downsample all decay datasets to have \( N_{\text{small}} \) events

- To avoid low training data, instead downsample all decay datasets to \( \text{upsamplingDegree} \times N_{\text{small}} \) events

- upsamplingDegree was varied between 1-10, optimal performance (AUC ROC) at 7
Additional Improvements on Classifiers

- Shuffling of training data helped to prevent overtraining on certain decay types.

- Re-scaling parameters improved performance, namely putting all $\chi^2$ input variables on a log scale before training.

- Improvements increased efficiency several percentage points each at constant trigger rate.
Performance of Improved Classifiers

**OneTrack**: eff = 28.1% at 1 MHz

**TwoTrack**: eff = 70.7% at 1 MHz

Note: lower efficiency modes are charm
Comparison with Cut-based classifiers

-Two-track classifier shows significant improvement over cut-based classifier

-One-track classifier shows slight improvement over cut-based classifier

-One-track cut-based classifier is a function fit to the response of an ML selection

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Classifier</th>
<th>Signal Detection Efficiency (at same FPR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Track</td>
<td>Cut-Based</td>
<td>21.4%</td>
</tr>
<tr>
<td></td>
<td>BDT</td>
<td>24.4%</td>
</tr>
<tr>
<td>2-Track</td>
<td>Cut-Based</td>
<td>60.4%</td>
</tr>
<tr>
<td></td>
<td>BDT</td>
<td>75.1%</td>
</tr>
</tbody>
</table>
Implementation of Models into Allen

-Late July and early August were spent writing C++ code to implement one-track and two-track models in Allen Pipeline for LHCb run 3

-Two corresponding new algorithms, one new sequence

-Secured an LHCb computing account, merge request coming soon

-Infrastructure to make implementing additional Catboost classifiers easier
Experimenting with 3-Track and 4-Track Triggers

- Classifier inputs are taken from LHCb Trigger Reoptimization paper†

- In run 2, these selections could only be done in HLT2. Allen tracking in run 3 is so fast that these selections can now be made in HLT1.

- 3-Track, 4-Track models trained separately, all D decays excluded

- Model hyperparameters were optimized by exploring parameter space

Analysis variables:

- n, mc, sum PT, vertex $x^2$
- $\eta$, $FD_{x^2}$, min PT,
- $IP_{x^2}$, $N$(tracks),
- $N$(tracks with $IP_{x^2} < 16$)
Performance of All Four BDT Models

Note: in actual analysis, use an OR selection on all of these lines to improve efficiency.
Summary of Work Completed

1. Optimized 1-track and 2-track BDT classifiers
2. Implemented optimal 1-track and 2-track models into Allen
3. Demonstrated efficiency of 3-track and 4-track classifiers

Next steps: -Submit merge request for new Allen algorithms
-Implement 3-track / 4-track into Allen pipeline