Topological Trigger Developments

Likhomanenko Tatiana*
“National Research Centre “Kurchatov Institute”, Yandex School of Data Analysis, Russia

LHCC Open Session, CERN, Switzerland, 4th March 2015

Topological Trigger

- Generic trigger for decays or beauty and charm hadrons
- Look for 2,3,4 track combinations in a wide mass range
- Designed to efficiently select decays with
- Use fast-track fit to improve signal efficiency and minbias rejection

Event is represented as set of SVR’s

```
SVR  SVR  SVR  SVR  SVR
```

**Good generated B** preselection

```
SVR  SVR  SVR  SVR
```

**other preselections**

```
SVR  SVR  SVR  SVR
```

```
SVR  SVR  SVR  SVR
```

Bonsai BDT vs Matrixnet

- HTL2 run1 – bonsai BDT (BBDT):
  - discretizes input variables before training which ensures a fast and robust implementation
  - converts decision trees to n-dimentional table to store
  - prediction operation takes one reading from this table

- HTL2 run2 – oblivious trees (Yandex MatrixNet):
  - features binarization before training process
  - construction on each iteration trees with the same depth
  - can be converted to the bonsai BDT

MC-2014: HLT2

Run2 vs Run1

```
Table 1. Comparison HLT2 relation to HLT1 between Run 1 and current models which use HLT1 track, HLT1 2-body SVR instead of previous HLT1 Run1. These channels are reconstructible signal decays with pt(B) > 2 GeV and tau(B) > 0.2 ps
```

MC-2015: HLT1 «1-track vs 2-body SVR», HLT2

```
Figure 1. LHCb trigger system
```

```
Figure 2. Signal events BDT efficiency dependence on background one. Here event prediction is defined as maximum of its SVR’s predictions
```

```
Figure 3. Scheme of event selection in topological trigger
```

```
Figure 4. The left picture is a model for bonsai tree, the right one is a model for oblivious tree
```

```
Figure 5. Comparison of Yandex MatrixNet models and HLT2 Run1 algorithm (BDT). For run2 models different preselections were used: truematch, range > 900, and without these selections
```

```
Figure 6. Inefficiency comparison of different output rates: 0.25%, 0.3%, 0.35%, 0.4%, 0.45% and 0.5%
```

```
Figure 7. Uboost-like model trained on truematch preselected data: efficiency vs btau for different WP
```

```
Figure 8. Uboost-like algorithms: flatten by btau variable or channel
```

```
Figure 9. TMVA Ada Boost and scikit-learn Ada Boost algorithms comparison on two training samples: preselected with truematch and without
```

```
Figure 10. Scheme of Run2 HLT1 and HLT2 triggers
```

```
Figure 11. Comparison of three models, trained on base features set (‘base’), base set with the addition (bitb. sumpt, sumipchi2) (‘add features’), and base set with training samples reweighting (‘base + reweight’). Also two output rates are used here: 0.25% and 0.4%
```

```
Table 1. Comparison HLT2 relation to HLT1 between Run 1 and current models which use HLT1 track, HLT1 2-body SVR instead of previous HLT1 Run1. These channels are reconstructible signal decays with pt(B) > 2 GeV and tau(B) > 0.2 ps
```

```
Figure 12. Comparison of HLT2 to HLT1 “1-track vs 2-body SVR” along with HLT2 gives 15-30% improvement over Run1. HLT2 will be much more efficient but how much more will depend on the output rate.
```

```
Figure 13. TMVA Ada Boost and scikit-learn Ada Boost algorithms comparison on two training samples: preselected with truematch and without
```

```
Figure 14.  Comparison of Yandex MatrixNet models and HLT2 Run1 algorithm (‘BDT’). For run2 models different preselections were used: truematch, minpt > 500, and without these selections
```

```
Figure 15. True Match Selection
```

```
Figure 16. TMVA Ada Boost and scikit-learn Ada Boost algorithms comparison on two training samples: preselected with truematch and without
```

```
Figure 17. Scheme of Run2 HLT1 and HLT2 triggers
```

```
Figure 18. Comparison of three models, trained on base features set (‘base’), base set with the addition (bitb. sumpt, sumipchi2) (‘add features’), and base set with training samples reweighting (‘base + reweight’). Also two output rates are used here: 0.25% and 0.4%
```

```
Figure 19. Comparison of three models, trained on base features set (‘base’), base set with the addition (bitb. sumpt, sumipchi2) (‘add features’), and base set with training samples reweighting (‘base + reweight’). Also two output rates are used here: 0.25% and 0.4%
```

```
Figure 20. Comparison of three models, trained on base features set (‘base’), base set with the addition (bitb. sumpt, sumipchi2) (‘add features’), and base set with training samples reweighting (‘base + reweight’). Also two output rates are used here: 0.25% and 0.4%
```

Conclusion

- Applying classifiers (MatrixNet) for HLT1 “1-track vs 2-body SVR” along with HLT2 gives 15-30% improvement over Run1.
- HLT2 will be much more efficient but how much more will depend on the output rate.

Participants: Philip Iren, Tatiana Likhomanenko (tatiana.likhomanenko@cern.ch), Egor Khairullin, Andrey Ustyuzhanin, Mike Williams

References:
- https://github.com/anaderi/lhcb_topo_trigger

`