Big Data & Machine Learning in HEP

Mike Williams
Department of Physics & Laboratory for Nuclear Science
Massachusetts Institute of Technology

March 17, 2016
The Large Hadron Collider

Big Data
Real-Time Analysis
ML
Propaganda

Outline

{ Big Data
Real-Time Analysis
ML
Propaganda }
Big Data & Real-Time Processing
Big Data

LHC detectors are BIG in both size and data output. E.g., ATLAS weighs over 7000 tonnes and has 100 million channels.

The beam-crossing rate at the LHC is 40 MHz, which results in each detector producing ~petabyte / second of data or zettabytes per year!
Big Data

Custom-built electronics used to read out the sensor arrays all perform zero suppression to drastically reduce the data rates.

By the HL-LHC era, each experiment will need to process in real time roughly 150 TB/s or 600 EB/year (comparable to largest-scale industrial data sets).
Real-Time Processing

Each experiment currently implements both hardware and software triggering stages (cascading approach). I’ll focus on LHCb (most advanced trigger).

Feature-building in custom electronics (e.g. FPGAs) used to greatly reduce rate, but online computing systems still have herculean tasks.
Real-Time Processing

27,000 physical CPU cores (>50k logical cores) are used to run the full offline reconstruction in real time.

10 PB of disk used for the buffering while final alignment/calibration done.

Final event selection done with access to best-quality data (mostly done during down time between fills).

Real-time reconstruction for all tracks with $p_T > 0.5$ GeV.

Data buffered on disk while alignment/calibration done.

Real-time reconstruction for all tracks, final PID, etc., all available to select events.
Real-Time Processing

In Run 3, LHCb will increase the post-zero-suppression data rate to 5 TB/s, but also remove the hardware trigger. The full reconstruction will be done in real time on all events.

ATLAS/CMS will go to 150 TB/s in Run 4, but for hermetic detectors it’s not feasible to go straight to software (wifi data transfer being investigated).
Offline Processing

The bulk of offline computing is done using the Worldwide LHC Computing Grid (WLCG) which has 170 computing centers around the world and runs over 2M jobs / day.

Currently, the LHC experiments persist about 30 PB / year of data -- but this still “isn’t enough”. Many strategies being explored (scouting, partial-event storage, data compression, etc) to increase the physics output.
Machine Learning in HEP
Two ML algorithms used in this stage that select about 70% of the output BW.

Real-time reconstruction for all tracks with $p_T > 0.5$ GeV.

Data buffered on disk while alignment/calibration done.

Real-time reconstruction for all tracks, final PID, etc., all available to select events.

Final event selection is currently roughly 30% ML based (first introduced into the main trigger algorithm for the start of 2011 data taking).

LHCb trigger using a custom-built ML algorithm.

Ghost Killers & PID

Ghost-killing neural networks do an excellent job of removing fake tracks (which is a big problem with the occupancies of the LHC).

Particle ID is now commonly done using neural networks as well. These combine the info from all subsystems and exploit the many (often obscure) correlations between features to ID particles.
When the LHC started, people were still worried about using ML. CMS actually published both cut-based and BDT-based Higgs exclusions for the WW decay mode -- so we can see directly what is gained:
When the LHC started, people were still worried about using ML. CMS actually published both cut-based and BDT-based Higgs exclusions for the WW decay mode -- so we can see directly what is gained:
Using BDTs

There are 3 standard uses for classification: (1) place a cut on the BDT response; (2) fit the BDT response; or (3) analyze the data in bins of BDT response.

For example, the recent LHCb pentaquark discovery analysis trained a BDT to identify the decay $\Lambda_b \rightarrow J/\psi p K$ then placed a cut on the BDT response for each candidate which gives the mass plot below. The BDT information was then not used further in the analysis (it just selects events to study).
Tag jets using a SV then feed properties to 2 BDTs. Fit the 2-D BDT response distribution in data to extract the b-jet, c-jet and light-parton-jet yields. Each BDT uses 10 features as input (ML as dimensional-reduction).

Most Higgs analyses now also work this way; i.e., they involve fitting BDT distributions. Essentially, one can reduce a large-N-D fitting problem into a 1-D fitting problem (while still using the N-D info). (left: VH(bb), 18 features input).
$\textbf{B}_s \rightarrow \mu\mu$

Calibrate BDT to have uniform response on signal, bin data in BDT response and analyze simultaneously.


Combining CMS&LHCb data gives $> 5\sigma$ and a BF $\sim 3\times 10^{-9}$ (consistent with the SM expected value).
ML is also now being used for regression. For example, CMS has moved to ML-based jet-energy corrections (e.g., for Higgs to bb).
Advanced Topics

• ML algorithms where the response is de-correlated from some set of user-defined features. See Stevens and Williams [1305.7248]; Rogozhnikova, Bukva, Gligorov, Ustyuzhanin, Williams [1410.4140]. Used to search for a new boson in PRL 115 (2015) 161802, and currently being used in many papers to come out soon.

• Deep learning is actively being explored as a way to remove the human-led feature-design stage. See Baldi, Sadowski, Whiteson [1402.4735].

• Parametrized classifiers to interpolate the optimal selection when only a discrete subset of possible signals has been simulated. See Baldi, Cranmer, Faucett, Sadowski, Whiteson [1601.077913].

• Hyper-parameter tuning using Bayesian global optimization.

• Blending classifiers, etc, etc ...
Tools

Most physicists are using either ROOT’s TMVA package or the python sklearn (sklearn for short). Some limited usage of proprietary packages too.

See also pypi.python.org/pypi/hep_ml/0.2.0 for some custom HEP algorithms.
Propaganda
The few of us (physicists) who do research in this area are convinced that direct collaboration/engagement with computer/data scientists is critical. We’d like to “open the box” to make it easier for those communities to do research on our (awesome) problems (please contact me if interested).
ML school for Ph.D. students in Lund (follows one last year in Moscow).

Go to bit.ly/mlhep2016, or email me, or mlhep2016@yandex.ru.
Summary

- HEP experiments are VERY big data. The online systems must process some of the world’s largest data sets.

- Amazingly, we’re now seeing real-time reconstruction, calibration, and alignment -- and soon we’ll see a triggerless-readout system at the LHC!

- Not long ago, using a ML algorithm was considered “unsafe”; however, now it’s rare to see a HEP data analysis that does not use ML algorithms. ML is used to kill fake tracks, do particle ID, classify events, regression, dimensional reduction, etc.

- ML algorithms are even starting to dominate real-time (trigger) selections.

- There are many great (free) tools out there that are pretty easy to use. It’s never been easier to get started. For students, there are dedicated summer schools to help. For everyone, there are now dedicated workshops.