Data Management and ML

Valentin Kuznetsov

S2I2 HEP/CS Workshop, Princeton 2017

Why should we care?

- Close the gap between HEP and CS/DataScience communities
- Gain expertise and novel ideas
- Establish collaboration and engage CS students to work on HEP problems
  - open access to HEP data

Current status

- When we deal with data we mostly oriented on production use case
  - data organization and movement by files or blocks
  - single data-format, ROOT not known outside HEP
- We complement data location by data discovery part
  - often experiments maintain data management metadata
- We process our data using sequential approach
  - RAW | GEN+SIM+DIGI -> RECO -> AOD -> MINIAOD
- Modern use cases require new way to handle the data

ML/DL example

- 90% of the time Data Scientists need to pivot/transform a data
  - data transformation
    - exploratory analysis; learn and create new features; apply dimensionality reduction
  - desire to use raw data (modern NN/DL frameworks will find out features from data)
- ML algorithms mostly use arrays or matrices rather then trees
  - it's easy to parallelize computation on flat data structures
- Most of the time ML deals with small datasets but not much on TB-PB ones

Routes

- Adopt ML algorithms for your data(-format), e.g. TMVA & ROOT
    - behind new ideas and innovations in ML world
- Adopt data (model) to existing and new ML tools
    - often require flatten trees; new data formats; what and how to deal with legacy data-formats
- Big engineering problem, CS+R&D
    - how to organize production pipeline: how to train ML on PB datasets, ML over distributed datasets, data transformation on a fly, single node, cluster, distributed clusters
    - dynamic dataset composition at large scale
    - best practices from big players and learn their expertise

We need

- ML front:
  - tools working with distributed with data
  - software -> specialized hardware
- Data management front:
  - data streaming: flexible and generic I/O system, e.g. read data via socket regardless of data location; random access to trees, events, branches, leafs
  - data transform on a fly
    - we can't afford another set of PB on disk for one data transformation
  - combine network access with storage/data-management layer, efficient caching+network access