RootInteractive expert tool for multidimensional statistical analysis, machine learning and analytical model validation.

Marian Ivanov (GSI Darmstadt), Marian Ivanov (UK Bratislava)
On behalf of ALICE collaboration

https://github.com/miranov25/RootInteractive
Alice Run 3 - goals and challenges

Record large pp and Pb-Pb minimum bias sample
- Continuous readout at 50 kHz Pb-Pb collisions and 500kHz-1Mhz pp collisions
- Unknown collision time
- Events overlapping in TPC → substantial higher occupancy (~5 PbPb collisions, 100 pp collisions)

Tracking challenge: space charge in TPC detector distorting trajectories
- Non-uniform space-charge distorting E field
- Large space point distortions $O(5 \text{ cm})$ and Distortion fluctuations $O(5 \%) \sim 0.2 \text{ cm}$
- To be calibrated to $\sigma \sim 100 \mu\text{m}$ with space granularity $O(10^6)$ in space $O(1-5 \text{ ms})$ in time

PID challenge: Significant baseline bias and fluctuation
- Online digital signal processing to recover baseline (in FPGA)
- To be corrected below internal noise level

A high interaction rate environment, pile-up, distortions fluctuation, etc. ... necessitates the use of advanced methods of data analysis. Experts and highly customisable tools are needed
RootInteractive project

Seeing is believing

Querying/Iterative
Interacting/predicting is understanding

- https://github.com/miranov25/RootInteractive#readme

Multi-Dimensional interactive analysis - ML, fits, histograming, data aggregation on server (Jupyter notebook, python scripts) and on clients $O(10^6-10^7$ rows, $10^8$ entries rows x columns) (browser)
RootInteractive - current ALICE expert projects

- Run3 alignment & space point distortion calibration
- Run3 digital signal processing
- Run2, Run3 track reconstruction optimization, validation
- MC/data mapping & TPC data volume studies
- Run2, (Run3) expert differential QA/QC, performance parameterization, performance web pages

- Run3 (4D) reconstruction development - trackCombinator - V0, Cascade, Kink, cosmic finder
- Fast simulation - fastMCalman for detector and reconstruction optimization (Run3,Alice3)
- Expert data representative sampling/skimming
- PID calibration/validation and dEdx optimization
- High dEdx, spallation tracking (collaboration with DUNE experiment)
- Magnetic monopole reconstruction

- Particle production - MC generators parameter scan
- Particle production as function of event properties
Multi-dimensional analysis vs shadow projections

Object and reference objects (models/reference models, MC/Data/Data/ref. data), should be compared optimally in the full relevant multi-dimensional space.

- Shadow projection → Assumptions, imagination and rhetorical art in describing data needed
- Comparison statements to be based on invariants or on normalized data - e.g. the difference between the object and the reference object
  - After projection impossible
- In many typical cases variance $\sigma_{A \ominus A_{ref}}$ is very often smaller by orders of magnitude
  - For example, the rms value of the difference between ionic currents and scaled average values can be used as an alarm criterion. We cannot use the ion current itself
- Differential approach - possibility to decompose and understand the data e.g. distortion due space charge, and alignment in figure above

$$\sigma_{A \ominus A_{ref}} \leq \sigma_{A^{(+)}} \sigma_{A_{ref}}$$

Track DCA bias due space charge distortion contribution before and after correction
Reference-ML prediction at low rate without SC
By oversimplifying in analysis level, the explanations tend to be more complex resp. wrong

Our goal to provide a tool to deal with multidimensional problem simplify data analysis in many dimensions:

- Fit and visualise N-dimensional functions including their uncertainties and biases
- **Easy to validate assumptions, numerically evaluate approximations, differentially compare models**
- Enable simple **functional composition** for (non-parametric, analytical/parametric) functions and error propagation
- Very fast feedback from day one - seconds instead of weeks, to allow interactive expert communication
- **Multidimensional parametric optimization**
- Easily configurable visualization of unbinned and binned data, interactive multidimensional histogramming/projection and derived **aggregate information extraction** on the server (Python/C++) and **client (Javascript)**
- **Client side application** (standalone HTML document) without necessity to install additional software

A detailed differential understanding of the detector system, MC and reconstruction/calibration performance is a prerequisite for the successful application of Machine learning in physical analysis
Consideration: symmetries, alarms and invariants

Aggregation/projections of normalized data e.g. (data-model), (MC-Data), (data-symmetry) in multiple dimensions:

- RMS spread is much smaller
- Alarms/Outlier tagging with statistical significance - e.g. (data-model) > N σ, or likelihood
- Invariance/symmetries
  - in-variance in time (using e.g. reference/average run), in-variance in space (e.g. rotation, mirror symmetry)
  - B field symmetry
  - data - non parametric/parametric analytical model
  - smoothness resp. local smoothness

In RootInteractive supported mostly comparing data with reference “symmetric regression” and “template support” automatic comparison to reference data
Multidimensional parameter optimization example - ALICE digital signal processing

Digital signal processing (13 parameters in example) needed for particle identification and data volume optimization. \( O(200000) \)

- Parameters: effects (On/Off), algorithm (different version), parameters of individual algorithms
- Simulation and visualization/aggregation (NDPipeline+RootInteractive) done by bachelor student, fully solving optimization problems of DSP (several attempts before failed)
- Dashboard to answer "all questions", FEEDBACK time for follow up questions \( O(\text{seconds}) \)
- Standalone dashboards, others could reproduce result based on the instruction in presentation, movie instruction
- Interactive expert use-case discussion within ONE meeting. DSP understand and solved. Project DONE.

Presentation, notebook, interactive dashboard and movie in RootInteractive tutorial:
- [https://gitlab.cern.ch/alice-tpc-offline/alice-tpc-notes/-/blob/master/JIRA/ATO-559/parameterScan.ipynb](https://gitlab.cern.ch/alice-tpc-offline/alice-tpc-notes/-/blob/master/JIRA/ATO-559/parameterScan.ipynb)
- [https://indico.cern.ch/event/1073883/contributions/4588170/attachments/2334149/3986420/simulScan_02112021.html](https://indico.cern.ch/event/1073883/contributions/4588170/attachments/2334149/3986420/simulScan_02112021.html)
- [https://indico.cern.ch/event/1135398/contributions/4764024/subcontributions/370740/attachments/2402507/4114272/CMITSimulationsGEMTPC.mp4](https://indico.cern.ch/event/1135398/contributions/4764024/subcontributions/370740/attachments/2402507/4114272/CMITSimulationsGEMTPC.mp4)
Machine learning in RootInteractive - differential validation of MC/data and ML models

Using external models:

- E.g comparing the U-Net for the distortion correction with simple data driven Machine learning using Random Forest
- Parameter optimization in respect to different cost functions

RootInteractive extensions wrappers to scikit-learn and xgboost

- Fast approximation of functions and local PDFs

Interactive validation in RootInteractive on client $O(10^6-10^7)$ points

- Unbinned predict
- Aggregated information for further postaggregation
  - Local mean, median, STD - unbinned predict
  - Local kernel regression parameters - aggregated information on the mesh
    - Usually statistical properties of predict- value, resp. Mash of 1D histograms
- Generalized kernel linear regression on client (ND groupby+rolling+kernel)
- Predict on client (wasm+ONNX) in queue
**Generalized linear (kernel) regression in RootInteractive - client side**

**Example, declaring generalized linear kernel regression**

```json
regressionArray=[
    {"name":"regre1", "varX":["x1","x2",...,"xn"], "varY":"y1", "weights":"w"},
    {"name":"regreAgg1", "varX":["x1","x2",...,"xn"], "varY":"y1", "weights":"w"}
]
```

- **Scikit-learn like user interface**
  - Using fit and predict
  - Regression predict new data source can be used as an alias function

- **Pol0 group-by regression, mean, median, quantiles, RMS**

- **Pandas groupby + ND-rolling/sliding kernel + Linear regression**
  - Interface as in the C++ code in original ND pipeline
  - Using fit and predict on the grid
  - Prediction of values and derived variables (using local fit parameters, e.g. local derivatives)
  - Predict is new data source
  - **Work in progress**

**Linear regression** is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables.

---

**Example of a cubic polynomial regression, which is a type of linear regression.** Although polynomial regression fits a nonlinear model to the data, as a statistical estimation problem it is linear, in the sense that the regression function $E(y \mid x)$ is linear in the unknown parameters that are estimated from the data. For this reason, polynomial regression is considered to be a special case of multiple linear regression.
Data preparation - RDataFrame <-> awkward (new interface)

**Defining RDataFrame**

```c++
ROOT::RDataFrame df(nTracks);
auto rdf = df.Define("qVector", "getQVector(160)"
    .Define("loggVector", "ROOT::VecOps::Log(qVector)"
    .Define("qStd", "StdDev(qVector)"
    .Define("qMean", "Mean(qVector)"
    .Define("qLStd", "StdDev(loggVector)"
    .Define("qUMean", "Mean(loggVector)"
    .Define("qMedian", "THash::Median(qVector.size(), qVector.data())"
    .Define("qUMedian", "THash::Median(qVector.size(), loggVector.data())"
    .Define("qTrunc", "truncate(qVector);"
    .Define("loggTrunc", "ROOT::VecOps::Log(qTrunc);";)
```

**Loading awkward array**

```c++
In [7]:
array = ak.from_rdataframe(
    rdf, columns="
        "loggTrunc",
        "loggVector",
        "qMean",
        "qMedian",
        "qStd",
        "qTrunc",
        "qVector",
        "qLStd",
        "qUMedian",
        "qULTrunc",
    ),
```

CPU times: user 1min 44s, sys: 0ms, total: 1min 45s
Wall time: 10.2 s

**dEdx optimization example**

- Defining the data and derived function (C++) with native data representation
- Loading the data → awkward array
- Execution scaling with number of cores (32 used in example)
- ML training/prediction → RDataFrame ()

*Significant performance increase with parallel "RDataFrame ↔ awkward" in respect to previously used direct Tree queries interface. Used extensively, e.g. in fastMCKalman (distortion simulation/correction) and in trackCombinator (V0,cascade,cosmic,loop finder) prototyping use case studies*
RootInteractive/Multi-Interactive project preparation and presentation

Expert data preparation

- Agreement on data to collect and aggregate
- Data sources
- Variables to import - asking questions
- Symmetries, invariances and possible alarms
- Pre-aggregation
- Data sampling
- Machine learning models
- Underlying Analytical models if exist
- Re-iteration

Data presentation:

- Agenda: presentation, notebook, dashboard+ (optional)movie
- Goal
- Data preparation explained
- Variables description
- Observation highlights with snapshot from dashboard
- Domain experts, participants in the meeting should be able to participate in decisions, resp. be able to interact with dashboard data based on description in presentation

The data is presented in a multidimensional way. The aim is to answer all questions within one meeting/session. If the information is not sufficient, new data sources to be agreed on.
RootInteractive pad map dashboard declarations

User defined RootInteractive properties are required to get the html output (explained in next slides)

- Alias array for derived variable/function definition - e.g defining status bitmask
  - aliasArray=[("IDC0_OK","0x2*(abs(IDC0_MeanRF0_LRatio)<sigmaRFCut0)) | (0x4*(abs(IDC0_MeanRFL_LRatio)<sigmaRFCutL))",..]
- Variable array
- Parameter array - to control parameterized functions, selection and variable selection for ND histograms
- Widget description array
- Widget layout dictionary
- Histogram array
- Figure array
- Figure layout dictionary

Simplification of using interface using set of predefined parameterizable templates to define standard layouts, extending only user defined widget control.

Templates focussed mostly on comparison of data and reference data, resp comparison of their distributions for user defined selection.
Functions on client - derived variables and functional composition

Predefined parametric javascript function

Custom javascript function (javascript function as a text)

Anonymous function (used for example in ND histograms as weights or variable)

Figure axis transformation

Many different ways to define derived variables and functional composition. Dependency trees to resolve functional and data source dependencies.
Histogram declaration - calibration QA browser

Set of the 2D, 3D (ND) histograms declared:

```
name: 'histogramData',
variables: ['varX', 'varY'],
axis: [1], 'quantiles': [0.25, 0.5, 0.75], 'unbinned_projections': True,
```

```
name: 'histogramData2',
variables: ['varX', 'varY', 'varZ'],
axis: [1], 'quantiles': [0.25, 0.5, 0.75], 'unbinned_projections': True,
```

```
name: 'histogramData3',
variables: ['varX', 'varY', 'varZ'],
axis: [1, 2], 'quantiles': [0.25, 0.5, 0.75], 'unbinned_projections': True,
```

QA example mean charge: left - raw values (varZ), right-normalized to "expectation" (varZNorm)

Anonymous function (used for example in ND histograms as weights or variables)

Parameterized histograms:
- Variables and weights could be any variable from data source (column, derived functions, anonymous function)
  - In the QA/calibration browser variables defined by user selecting (varX, varY, varZ)
  - Binning controlled by parameters (nbinsX, ...)
- Derived aggregated data exported as new data source
  - Declaring quantiles and projections
  - Projection could be binned (fast) and unbinned

Customizable Ndimensional histograms and projection. Example:
- X, y median profile of cluster charge map (left) and normalized to phi symmetric RF prediction
Webasm interface - under development

New functions/transformations/data sources using wasm:

- Fast Fourier transform
- Convolution, deconvolution
  - Numpy like interface (binned data)
  - Functional interface (unbinned kernel function)
- ONNX interface
- Based on benchmark transformation of the older javascript numerical code to wasm

https://webassembly.org/
RootInteractive - conclusion

RootInteractive is used extensively and successfully in many ALICE use cases for multidimensional analysis

- Most important expert tools for the many use cases - e.g. distortion calibration, reconstruction and dEdx optimisation, preparation of a new reconstruction algorithm (trackCombiantor)

Current use cases, now mainly related to detector (calibration, simulation, QA) and global reconstruction (RUN3, RUN2 as reference, Alice 3)

Pilot N-dimensional physical analysis with sampled/skimmed data is in the queue
Backup
DCA-DCA0 bias - rate evolution (4,330 kHz, 660 kHz)

DCA bias in phi direction strongly eliminated - residuals O(2 mm) **comparable with intrinsic resolution** of the tracks in vertex O(0.2 cm). New analytical fits - fitting also density profile

DCA bias in theta direction strongly eliminated. Remaining bias due charge up on C side - to add up in the analytical fit version (IFC and OFC fit). Charging up rate and time dependent (see Run1, Run2 studies)
RootInteractive usage in ALICE

in following slides code snippet with user code declaration shown for illustration without further discussion
Machine learning - derived variables - RF regression - per channel QA example

### Defining models:
- varying parameter of models, input variables and local statistics

**Global (varListG)** and **local regression (varListLocal)** extracting for basic calibration and QA properties of ALICE TPC calibration and QA variables

- **global φ symmetric** model, local model **without φ symmetry**
- **Automatic alarms - data “out of range”** | **data-prediction | <nσ** without “reason” (other calibration, masking known problems)

Robust local statistics - median and local std estimator for the outlier tagging and PDF description