New and upcoming features in TMVA

Lorenzo Moneta (CERN)
Omar Zapata (Metropolitan Institute Of Technology & University of Antioquia)
Sergei Gleyzer (University of Florida & CERN)

outline

ROOT
Data Analysis Framework

TMVA

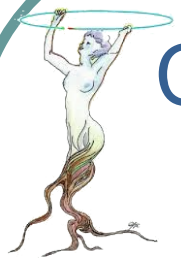
- Present
  - Current machine learning tools in ROOT
  - DataLoader
  - PyMVA
  - RMVA
  - Variable Importance
  - Cross validation
  - Deep Learning
- In Progress or Future
  - TMVA refactoring
  - TMVAGui
  - ROOTBooks
  - Cross validation
  - Deep learning
  - Parallelization
  - Memory and Test Suite
  - Additional features

Future of TMVA document

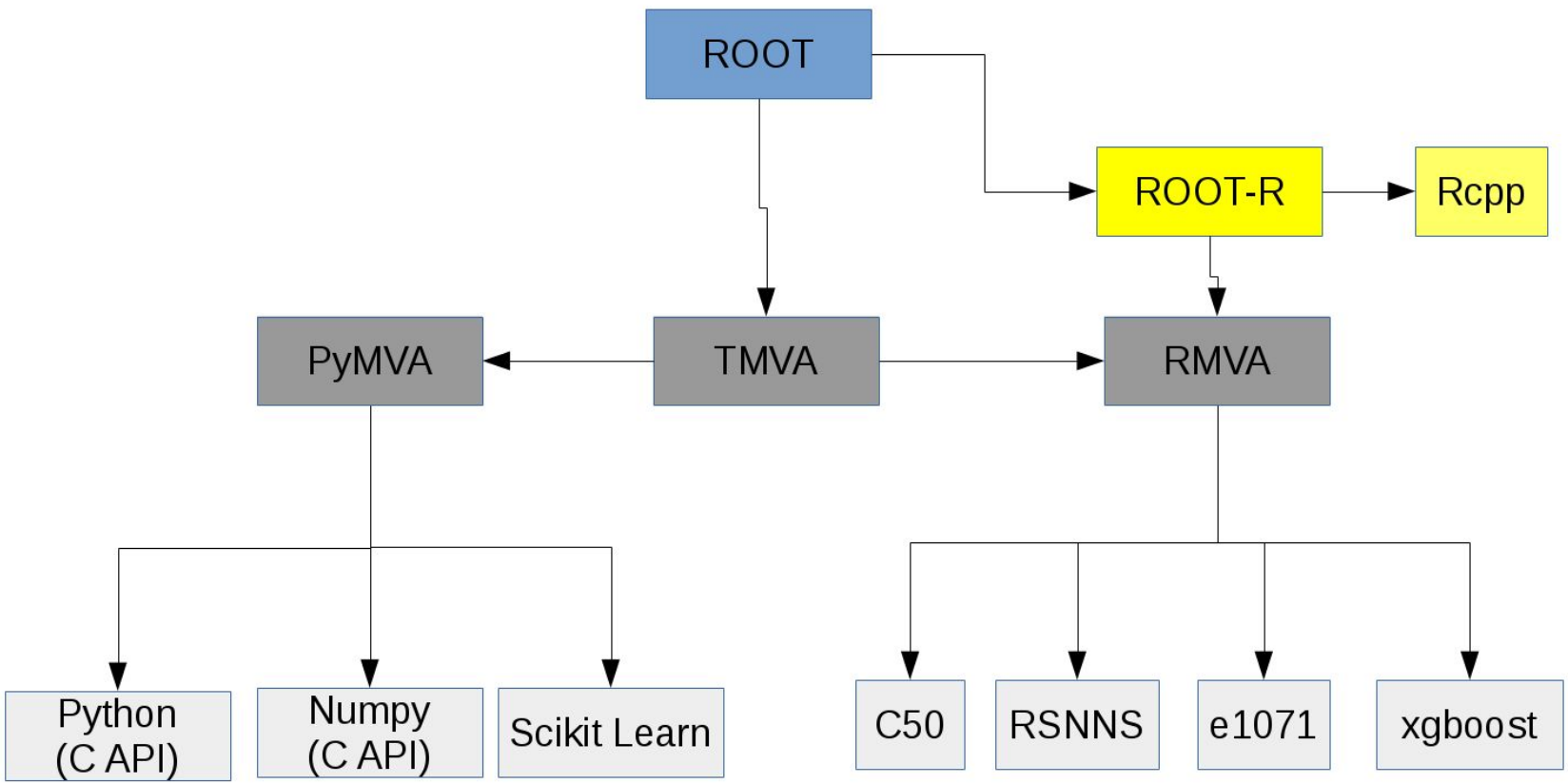
- Flexibility
  - The code should be made more modular **In Progress**
  - The core should be more flexible, allowing for decoupling for datasets/methods/variables **Done**
  - Pause and resume training **Not started**
- Core
  - Interfaces for R and Python **Done**
  - GPU support **Not started**
- Computational Performance
  - Core redesigned for improved computational performance **In Progress**
  - Dataset I/O should be revisited
    - Alternative input file types (Example: HDF5) **Not started**
- Desired Features
  - Cross -validation **In Progress**
  - Additional information for Analyzer (Variable importance) **Done**
  - Parallelization **In Progress**

see (Draft version 16/09/2015) http://iml.cern.ch/tiki-download_file.php?fileId=1

Present

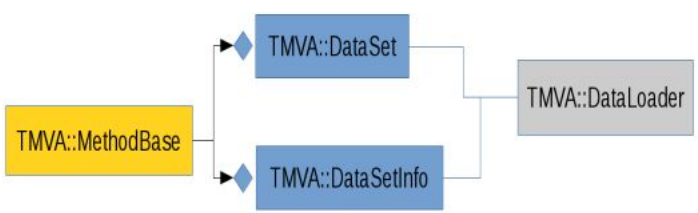


Current Machine Learning Tools in ROOT



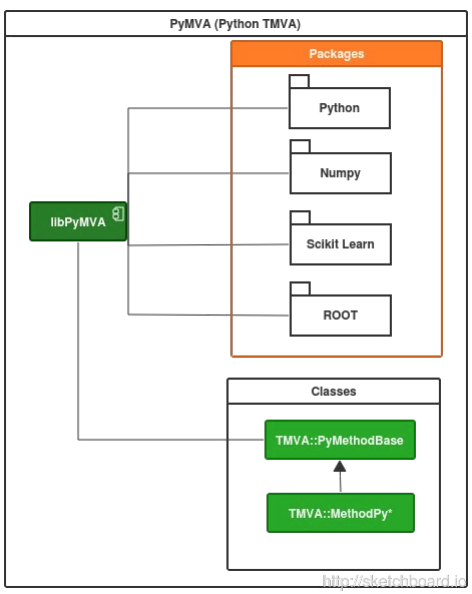
DataLoader

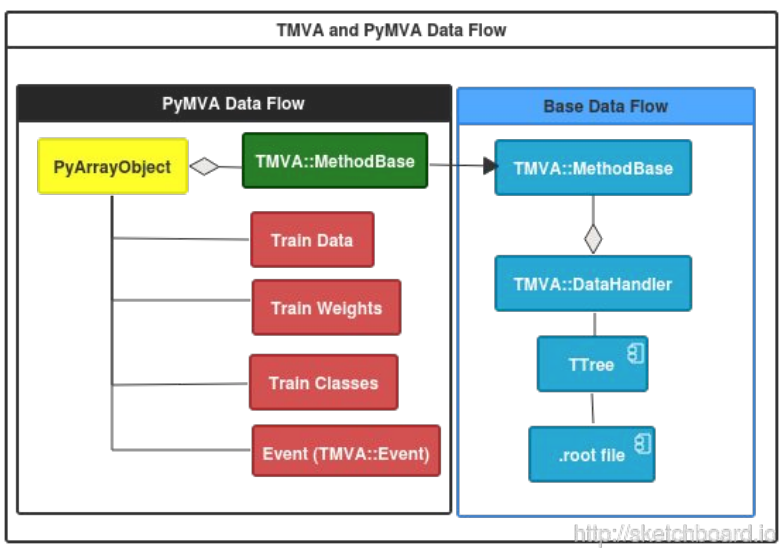
- TMVA DataLoader is a class that allows greater flexibility in working with datasets
  - Connects features with classifiers
  - Works with TTrees but extendable to other types
    - data frames
    - vectors



PyMVA

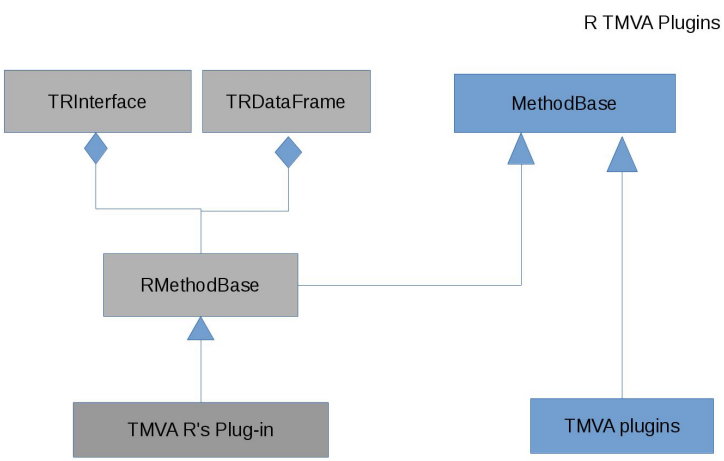
PyMVA is a set of TMVA plugins based on python api that allows new methods of classification and regression calling Python's packages.

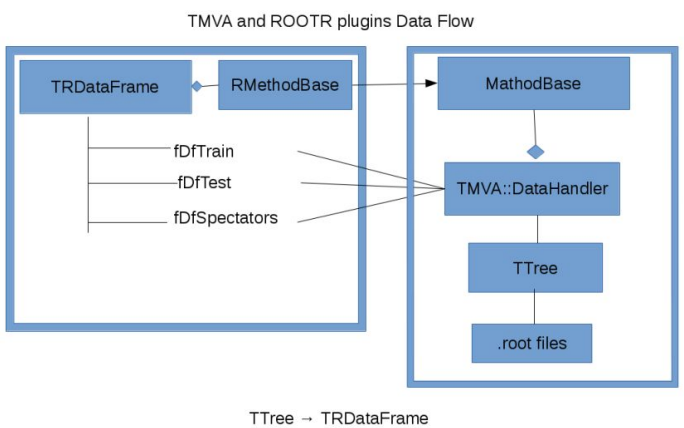




RMVA

RMVA is a set of plugins for TMVA package based on ROOTR that allows new methods of classification and regression calling R's packages.



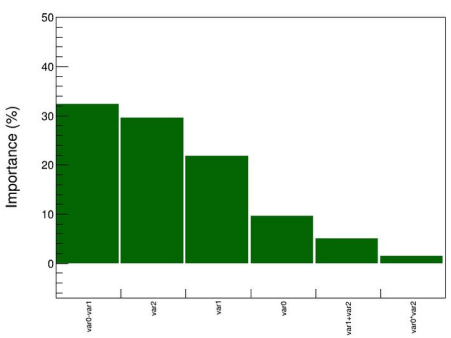


Feature Importance

- Ranks the importance of features based on contribution to classifier performance

$$FI(X_i) = \sum_{S \subseteq V : X_i \in S} F(S) \times W_{X_i}(S) \qquad W_{X_i}(S) \equiv 1 - \frac{F(S - \{X_i\})}{F(S)}$$

  - Feature set {V}
  - Feature subset {S}
  - Classifier Performance F(S)



- A stochastic algorithm independent of classifier choice

Cross Validation

- Currently performed with external scripts with TMVA
- Implemented by Thomas Stevenson and Adrian Bevan (see ACAT 2016 talk)

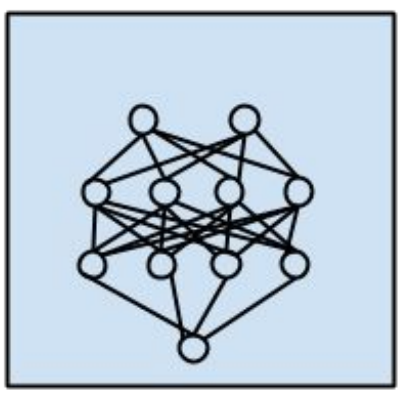
https://indico.cern.ch/event/397113/session/16/contribution/147/attachments/1214583/1773051/SVM_and_Generalisation_ACAT.pdf

Deep Learning

- Deep learning classes (plugin) in git branch https://github.com/root-mirror/root/pull/104

- Contains recent developments in the field
  - Weight initialization [Gerlot]
  - SGD
  - Hogwild style multithreading
  - drop-out
  - momentum
  - Multithreaded training

- Written by Peter Speckmayer
  - Undergoing final tests/evaluation.



Deep Learning Algorithms

SVM

Additional functionality for SVMs includes:

- Multi-gaussian, product and sum kernel functions added, as well as polynomial kernel function re-enabled.
- Parameter optimisation for kernel parameters and cost added following the implementation for BDT parameter optimisation.
- Can specify the parameters to be optimised and the range over which they are optimised.
- Calculation of loss functions, though not currently used.
- Weighting of the cost parameter to the relative signal and background dataset sizes, as to not bias the SVM training if one dataset is significantly larger than the other.
- Also return of map of optimised parameters from the tmva factory to allow for use in external program.

see next talk by Thomas
https://indico.cern.ch/event/397113/session/16/contribution/147/attachments/1214583/1773051/SVM_and_Generalisation_ACAT.pdf

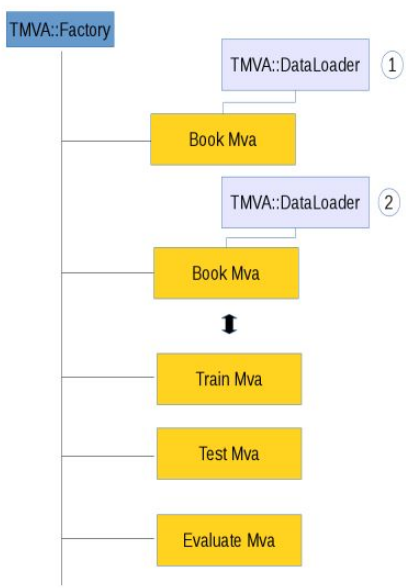
Features In Progress

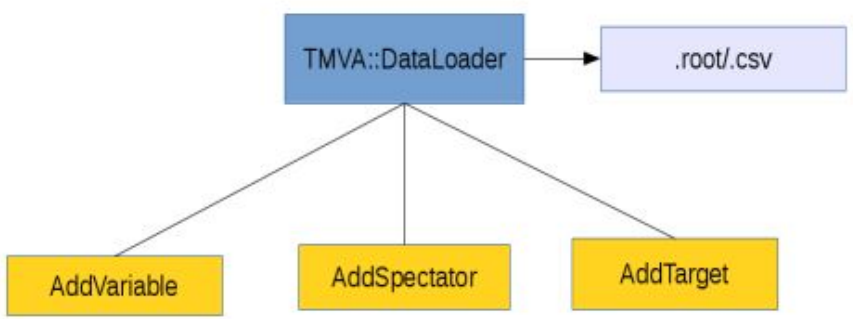
Upcoming TMVA

- Currently TMVA is undergoing important updates for incoming features
  - Removal of static variables from the code to support parallelization and avoid concurrency problems.
  - Creation of lightweight constructor that does not save results in ROOT file.
  - Separation of classification and regression classes
  - Faster classification and regression and statistical experiments like cross validation, parameters tuning, dimensionality reduction, hyperparameter optimization, etc..

DataLoader

- TMVA DataLoader will support reading .csv, HDF5, JSON, Custom Serialization FIles and SQL.






```
factory->BookMethod( DataLoader &,Types::EMVA ,  TString methodTitle, TString theOption);
```

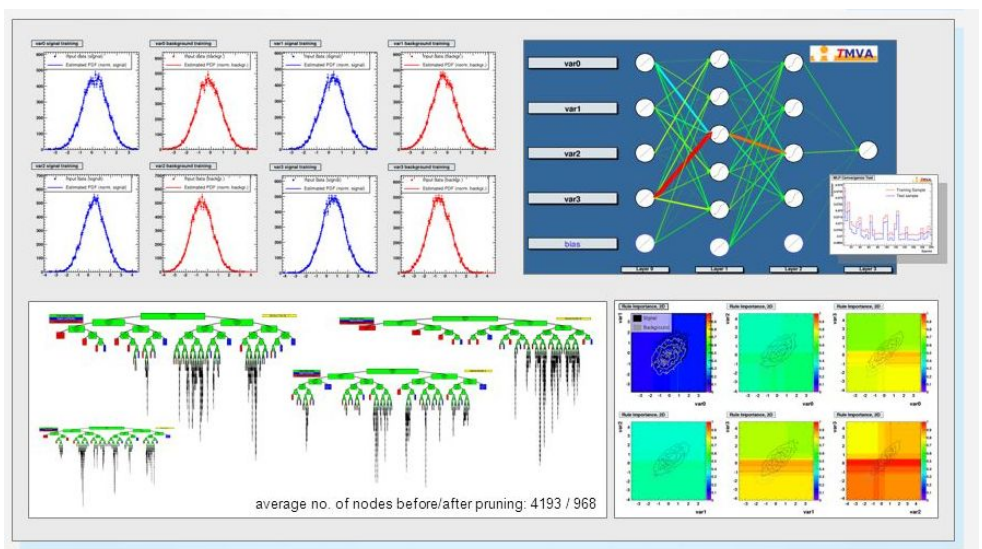
ROOT Books

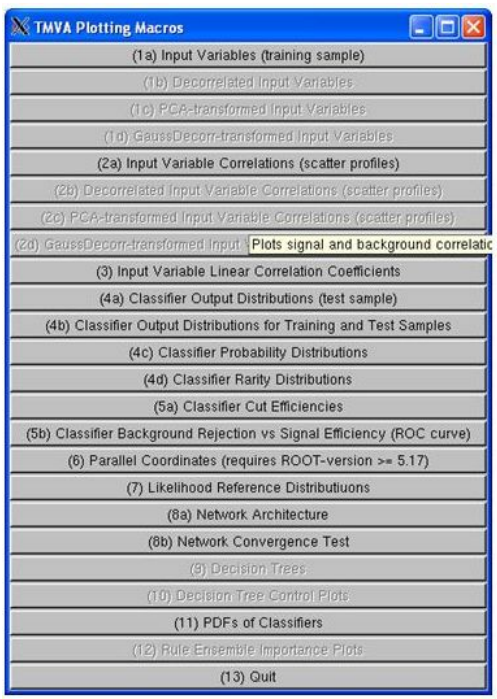
- Additional integration with ROOTBooks
  - ROC plots
  - Classifier structure visualizations
  - Plots on demand
  - Python support

TMVAGui

In the new design TMVAGui also needs to be updated:

- Integration with ROOTBooks
- Visualization of multiple datasets

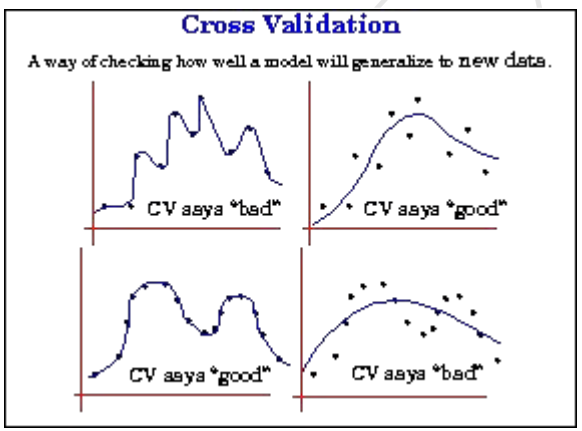




Cross Validation

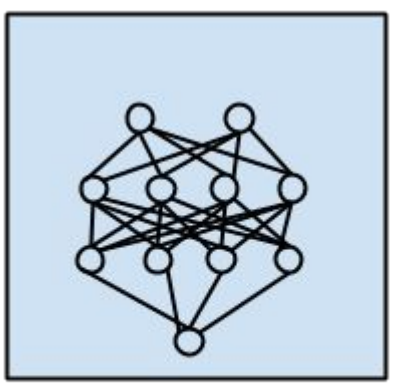
Fully integrated into TMVA

- New class TMVA::CrossValidation
- Will support parallel execution
- Optional hyper parameter tuning.



Deep Learning

- Additional deep learning plugins
  - packages with GPU support like darch
    - using PyMVA and RMVA.
  - restricted boltzmann machines



Deep Learning
Algorithms

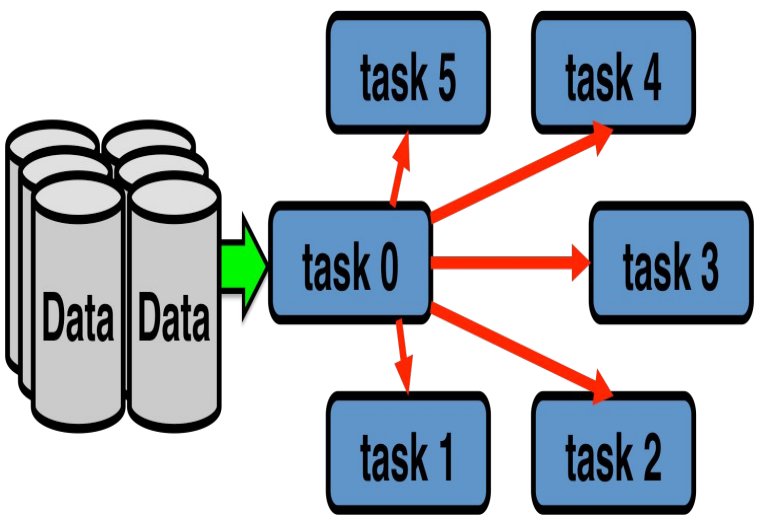
Parallelization

Parallelize TMVA using different approaches.

- Parallelize multiple methods booked into factory when Train/Test is called.
- Internally parallelize methods.

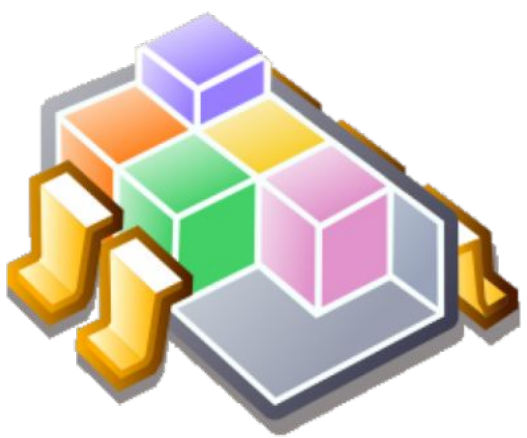
Technologies to be implemented

- ROOT MultiProc
- Threads and GPU support.



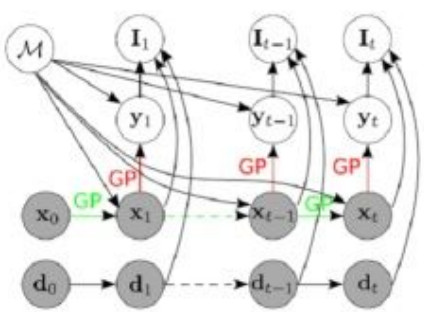
Memory and Tests Suite

- Memory optimization
  - Manipulation of multiple datasets
  - Data from multiple file types
  - Valgrind scripts for memory leaks detection
  - Well implemented memory use in parallelization:
    - SIMD (Single instruction, multiple data)
    - MISD (multiple instruction, single data)
    - MIMD (multiple instruction, multiple data)
- Classes for tests suite integrated to ROOT
  - Using CppUnit
  - Using macros for stress tests.



Additional features

- Gaussian processes for machine learning.
  http://www.gaussianprocess.org/gpml/
- Reduced support vector machines
- Parallel random grid search.
- Update documentation with doxygen.
- XML converter (see talk by Iurii)



doxygen

XML

More Information

Websites

http://root.cern.ch
http://oproject.org
http://iml.cern.ch

Thanks

ROOT

Data Analysis Framework