Overview of Recent Developments in ROOT/TMVA

S. Gleyzer (University of Florida), Lorenzo Moneta (CERN),
O. Zapata (Metropolitan Institute of Technology & University of Antioquia)

HSF Workshop, Orsay Paris, 2-4 May 2016

Outline

- Introduction
- Present status of TMVA
- New tools added last year
- Features added recently
- Overview of current progress
- Future planned improvements
- Conclusions

Introduction

- Community effort to improve ML tools in HEP
- Identified area of improvements
  - Inter-experimental Machine Learning working group
    - with participation of CERN SFT
    - endorsed by all LHC experiments
      - *see following IML presentation by Sergei and Steven*
- New developments happening recently in ROOT / TMVA are resulting from this effort

Document on the Future of TMVA

- Meeting in September to discuss future of TMVA.
- Written a draft document
  - see http://iml.cern.ch/tiki-download_file.php?fileId=1
- Core Requirements
  - maintain a set of core algorithms for HEP standard usage.
  - Interface to R and Python for high performance use (to allow using modern ML packages) **Done**
  - Facilitate workflow with external packages (e.g. DNN packages)
    - external training and apply their results in TMVA **In progress**
    - support exporting of input ROOT data to external packages and importing their results in TMVA **In progress**

Requirements for TMVA

- Flexibility
  - re-design for more modularity and for decoupling datasets/methods/variables **Done**
- Computation Performances
  - improve algorithms performance by optimising code, using vectorization and parallelisation **In progress**
  - Revised DataSet I/O **Not started**
  - optimising memory usage **In progress**
- Desired New Features
  - Cross Validation **Done**
  - Hyper-parameter tuning **In progress**
  - Additional Information for Analyser (Feature Importance) **Done**
  - Parallelisation and GPU support **In progress**
  - Support for alternative input files (e.g. HDF5) **Not started**
  - Pause and Resume Training **Started**

New ML Tools added in TMVA

- Overview of tools added recently in ROOT/TMVA
  - Last Year
    - DataLoader
    - Interface to Scikit-Lear (PyMVA)
    - Interface to R (RMVA)
    - Feature Importance
  - This Year
    - Deep Neural Network,
    - Improved SVM
    - Cross Validation and hyper-parameter tuning

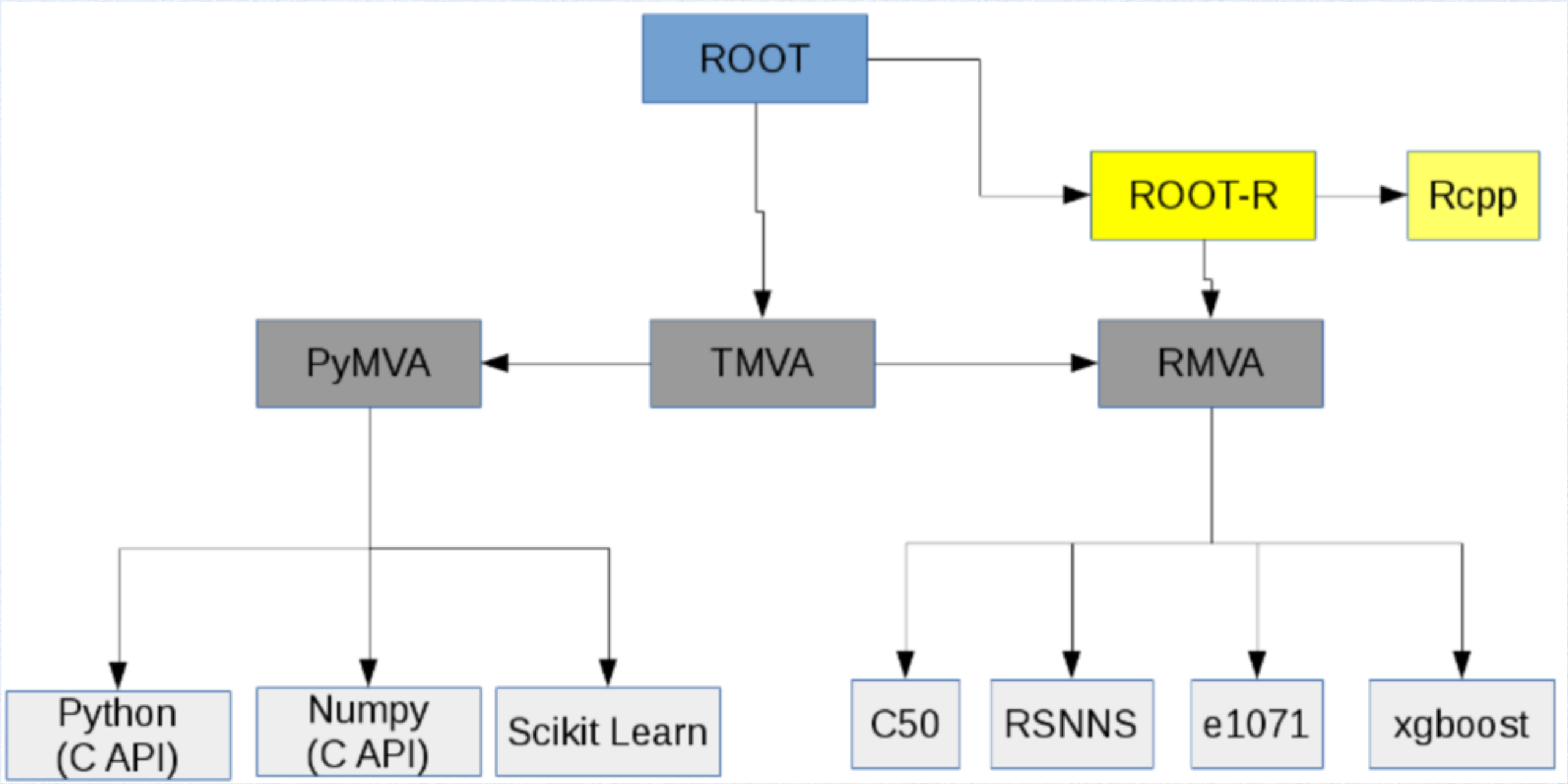
TMVA DataLoader

- **DataLoader** is a new class that allows greater flexibility when working with datasets. It is an interface to
  - load the datasets
    - root files (TTrees) but can be extended to other types (e.g. CSV, HDFS)
  - add variables
- TMVA Factory links **DataLoader** with a specific MVA method when booking

```
factory->BookMethod( DataLoader *loader, Types::EMVA theMethod,
                     const char * methodTitle, const char *option = "" );
```

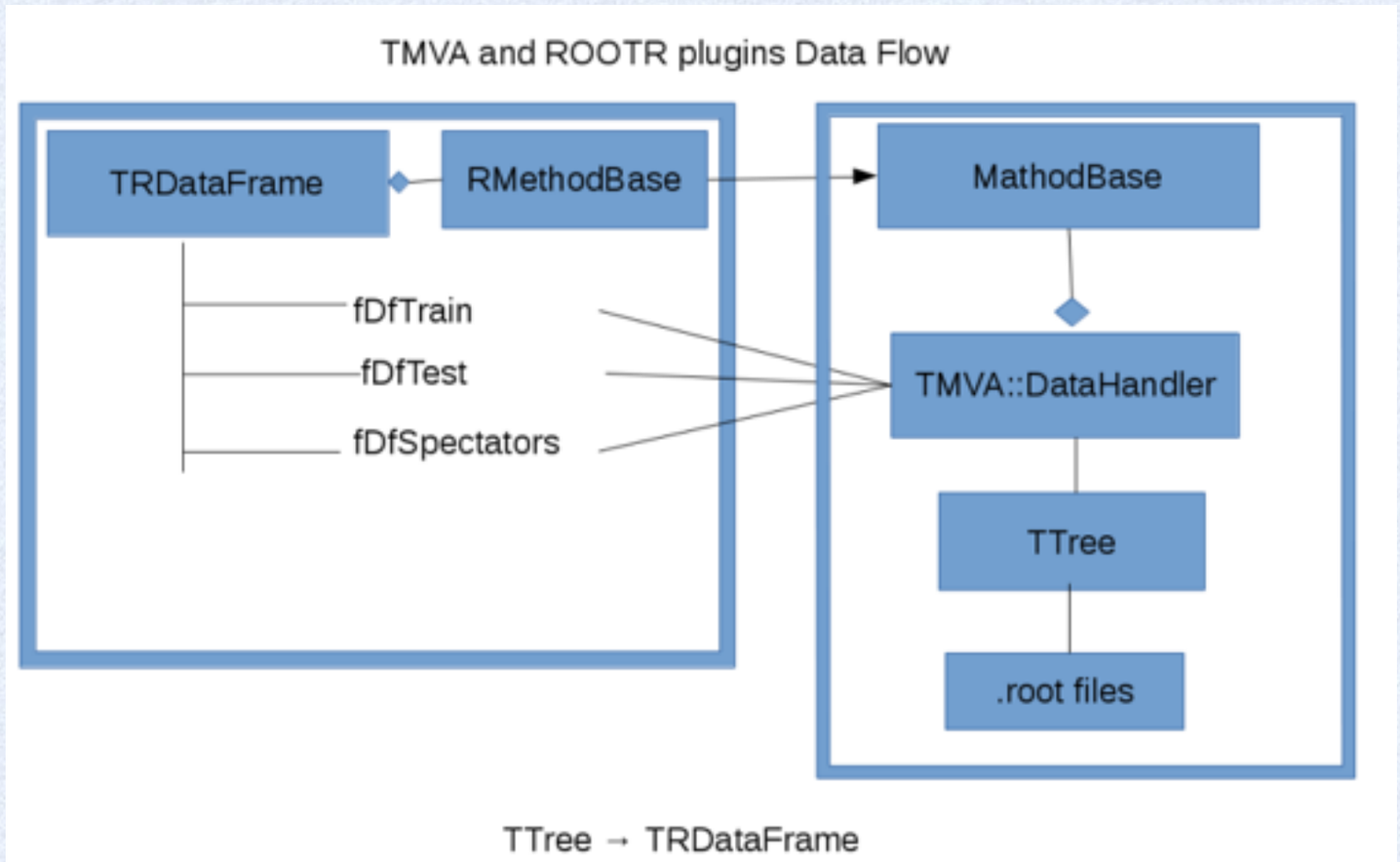
- Obtained desired flexibility in de-coupling methods/dataset/variables

Interfaces to R and Python



R-TMVA

- Interface R methods for Machine Learning in TMVA
  - use new ROOT-R package (allows to use R within ROOT)
  - set of plugins for TMVA based on R packages for regression and classification
  - available methods: **C50, SVM(e1071), RSNNS, XgBoost**

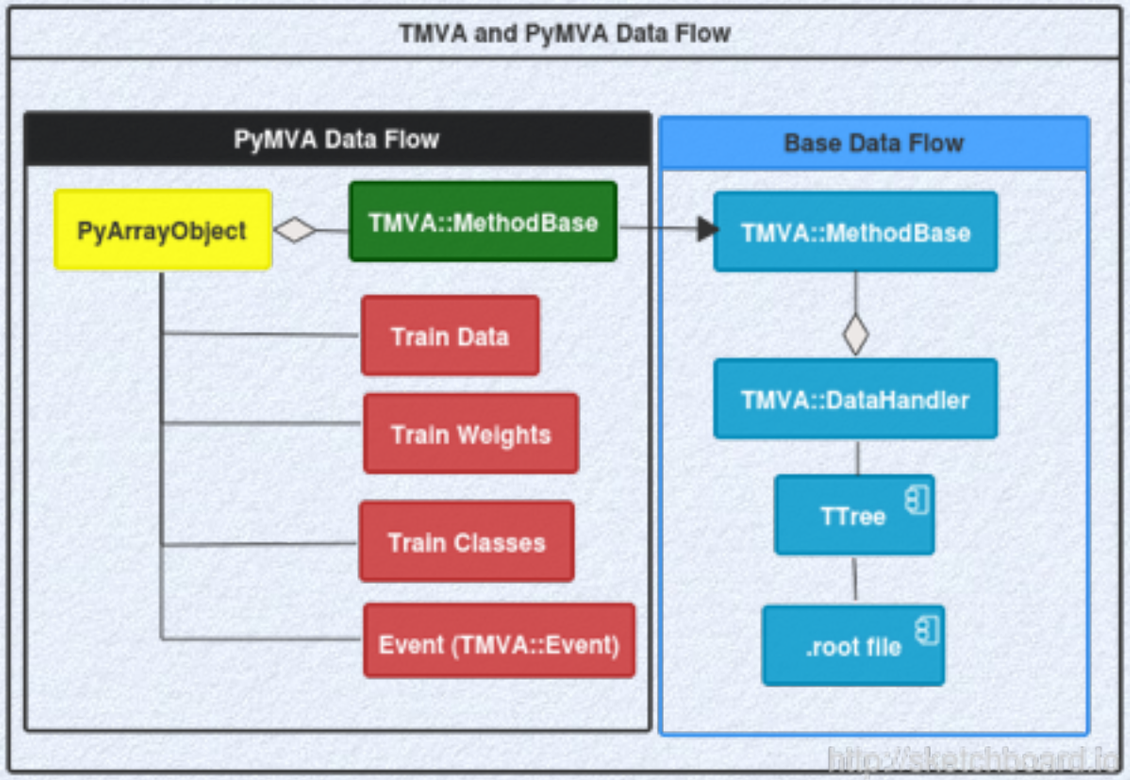


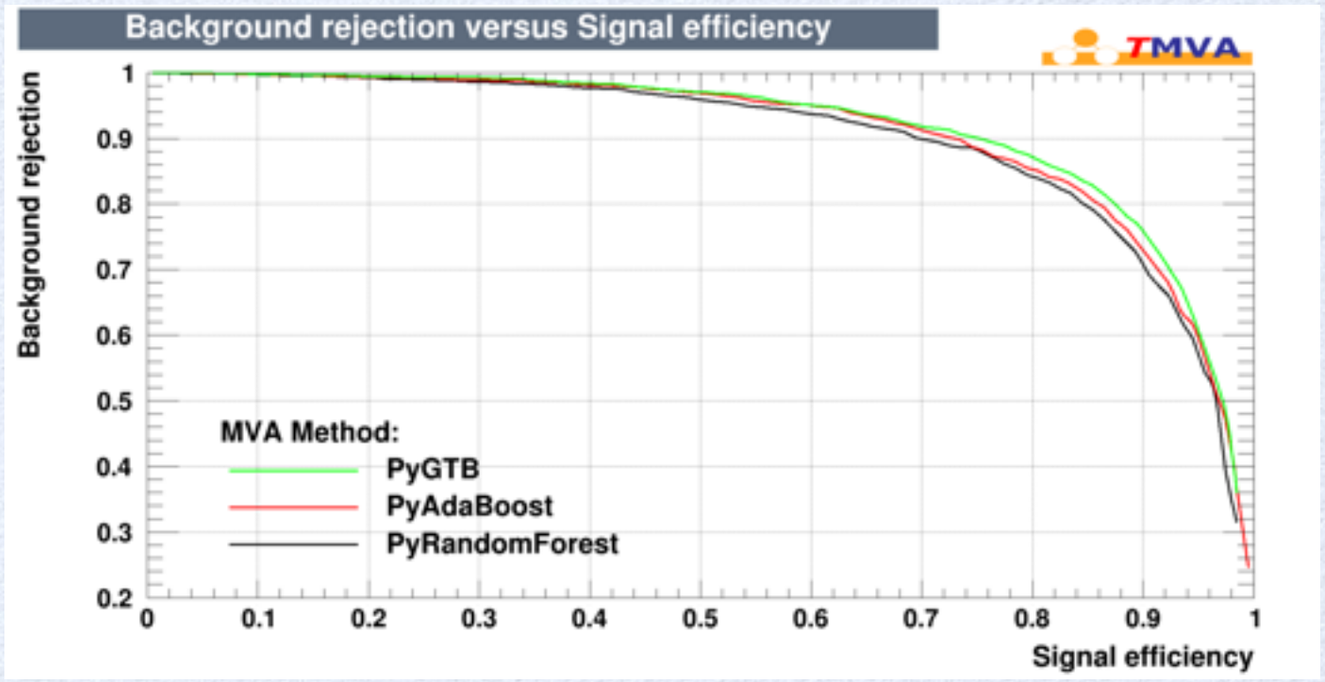
- Map ROOT data in a R data frame (TRDataFrame)
- Implement new R methods as derived class of TMVA::MethodBase

Available from ROOT 6.05.02. See doc at http://oproject.org/tiki-index.php?page=RMVA

PyMVA

- Interface to use Python ML tools from TMVA
  - Use methods from **Scikit-Learn** package
    - **Random Forest, Gradient Tree Boost, Ada Boost**
  - Convert input ROOT data in PyArrayObjects (C interface to numpy)
  - Use directly Python from C++ using its C interface





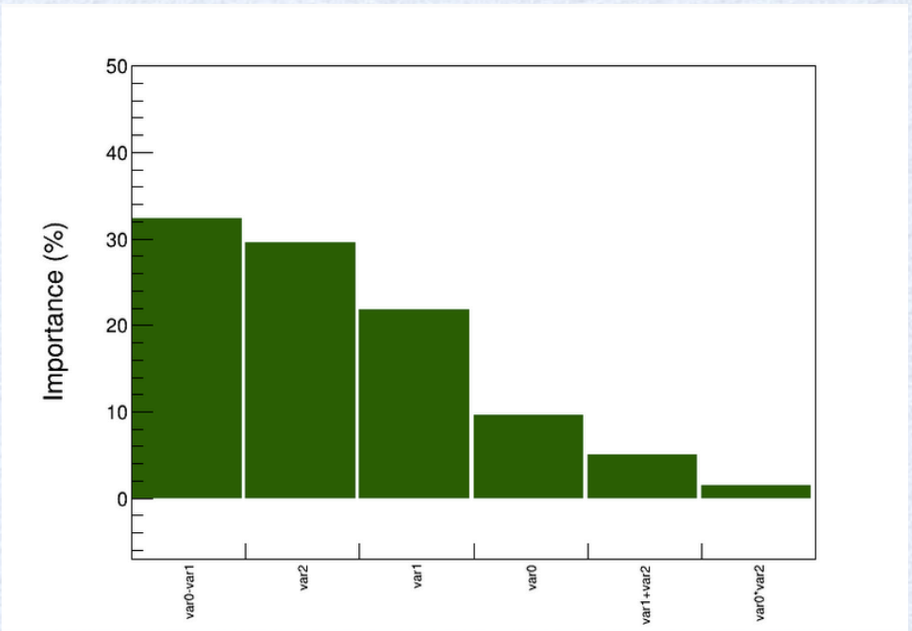
see http://oproject.org/tiki-index.php?page=PyMVA code available from ROOT 6.05.02 !

Feature Importance

- Ranks the importance of features based on contribution to classifier performance
  - A stochastic algorithm independent of classifier choice

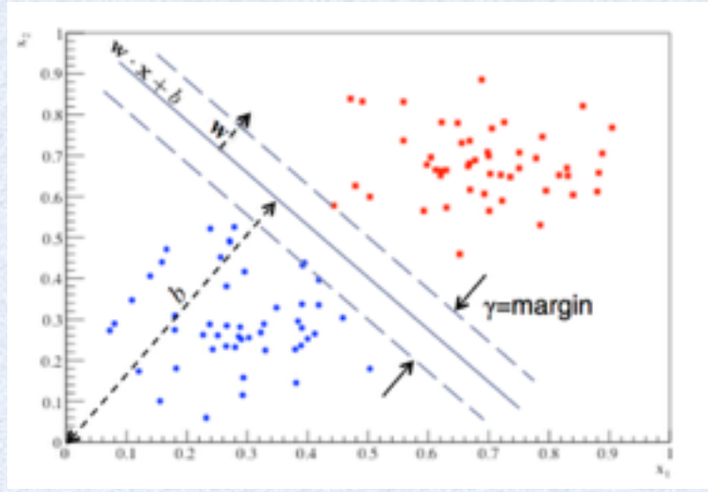
$$FI(X_i) = \sum_{S \subseteq V : X_i \in S} F(S) \times W_{X_i}(S) \qquad W_{X_i}(S) \equiv 1 - \frac{F(S - \{X_i\})}{F(S)}$$

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



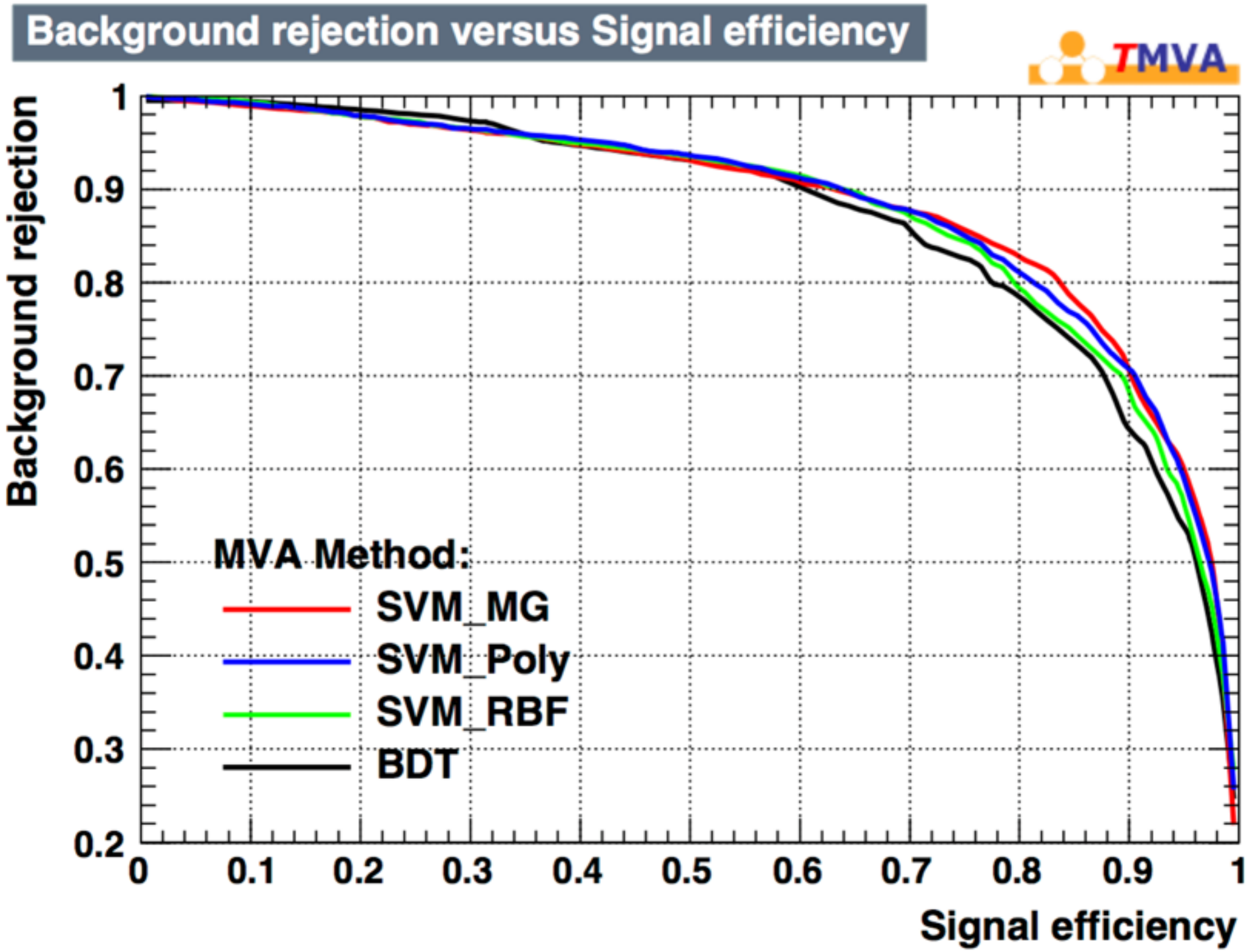
Improved SVM

- Additional functionality for SVM included in TMVA (work by *T. Stevenson* and *A. Bevan*)
  - New Kernel functions:
    - Multi-Gaussian, Polynomial and support for product and sum of kernel functions



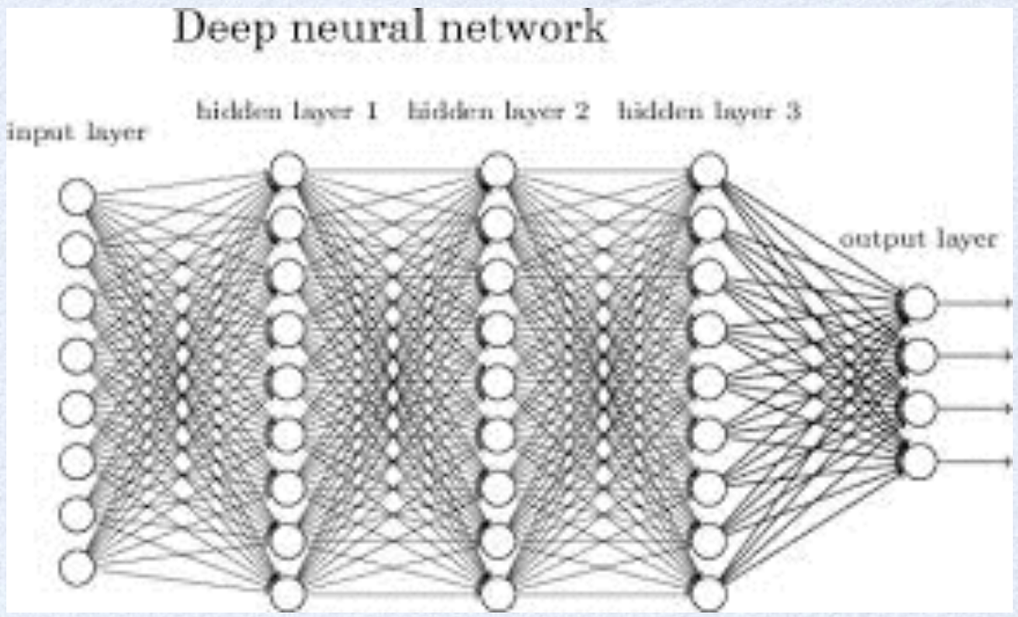
  - Implemented Parameter optimisation for kernel parameters and cost
    - Cost weighted to signal/background events
  - Loss function (implemented but not currently used)

EXAMPLES – HIGGS ML CHALLENGE DATASET



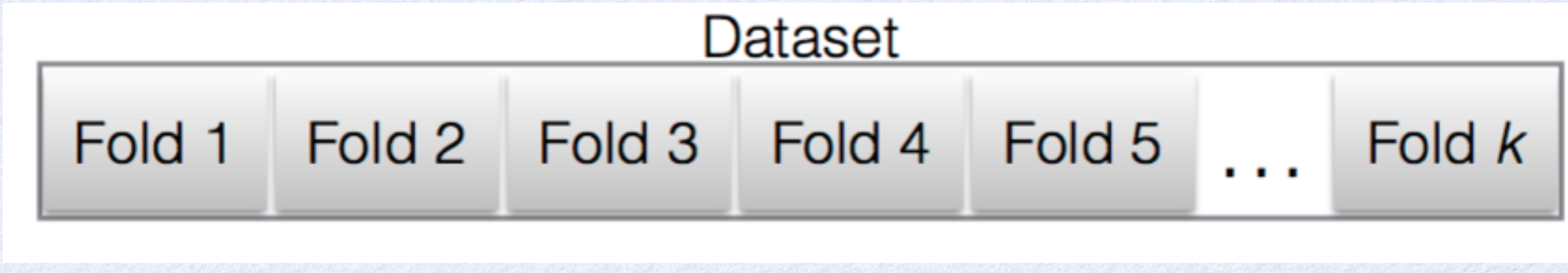
Deep Learning

- New Deep Learning classes added recently in TMVA (ROOT master version)
  - originally written by *P. Speckmayer*
  - optimisation in progress by TMVA developers
- Contains some recent developments in the field
  - Stochastic Gradient Descent (SGD)
  - Multithreading training support
  - Weight initialisation
  - drop-out
  - momentum



Cross Validation

- k-fold cross-validation



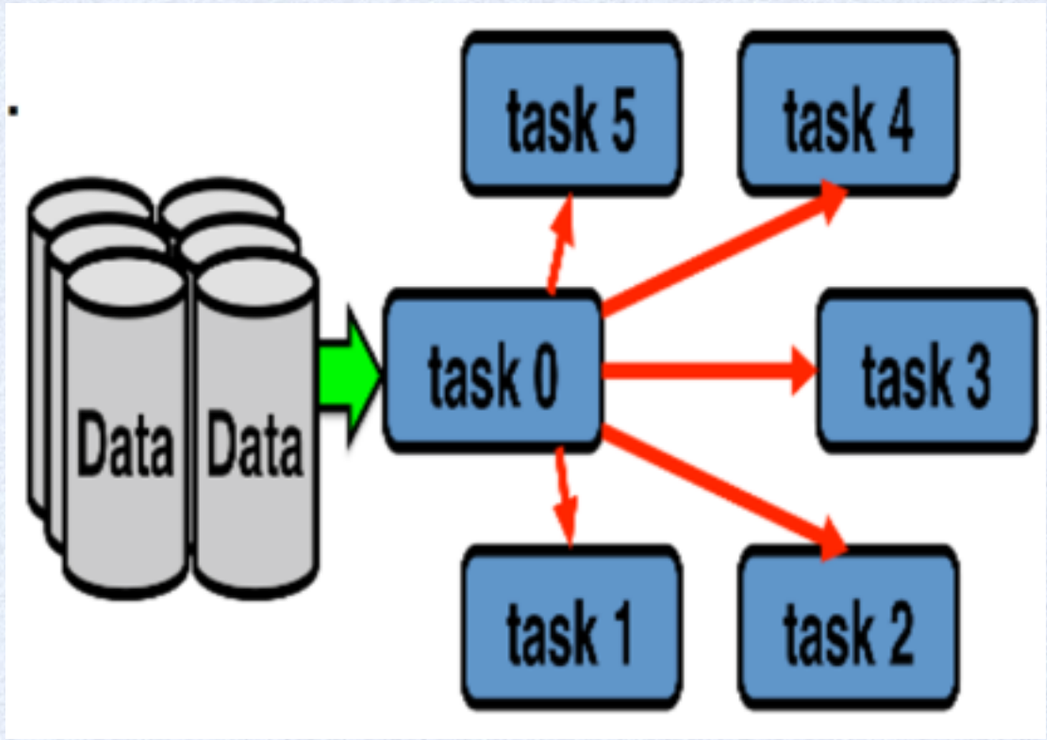
  - with optional hyper parameter tuning
- Implemented as standalone version by *T. Stevenson*
- Integrated now into TMVA
  - soon with support for parallel execution (Spark and multi-processes)

Upcoming New Features

- Improvements currently undergoing in TMVA :
  - Better separation of classification and regression classes
  - Improve regression
    - e.g. add option for different loss functions
  - Improve performance and memory usage
    - optimised code, usage of SIMD vectorisation, etc…
  - Greater support for parallelisation
    - removal of static variable to avoid concurrency problems

Parallelization

- On-going parallelization work:
  - Parallelise multiple methods booked into the factory when training and testing
  - Parallelization of cross validation and hyper-parameter tuning
  - Internal parallelization of methods whenever possible
- Using Technologies:
  - ROOT MultiProcess using fork (TMultiProc)
  - Multi-Threads using tbb (ThreadPool)
  - Cluster parallelisation using Spark
  - GPU



ROOT-Book Integration

- Additional integration with Jupiter notebooks (ROOT-Books)
  - ROC plots (already done)
  - Classifier structure visualisation
  - Plots on demand and integration with TMVA GUI
  - Python support
- Useful for interactive analysis
  - e.g. using **SWAN**: **S**ervice for **W**eb based **An**alysis

SWAN

SWAN: Service for **W**eb based **An**alysis

- Platform independent: only with a web browser
  - Analyse data via Jupyter Notebook web interface
  - No need to install and configure software
- Integrated in CERN services' portfolio
  - Calculations "in the cloud"
  - Allow easy sharing of scientific results: plots, data, code (EOS, CERNbox)
  - Simplify teaching of data processing and programming
  - Eases analysis reproducibility
  - C++, Python and other languages or analysis "ecosystems"
    - Interfaced to ROOT, TMVA, R…

swan.web.cern.ch

Future Improvements in TMVA

- Persistency of methods
  - use general ROOT I/O (and not be limited to XML) for output of training
  - import output from training performed from external packages (e.g. Scikit, Theano, etc..)
- Data Input
  - support for different input data sets (e.g. HDF5)
  - improve data handling classes in TMVA to avoid copying all data in memory

New GSOC Projects in TMVA

- 5 students this summer supported by Google (Google Summer of Code program) working on ROOT Machine Learning tools
  - Improvement of pre-processing layer
  - Parallelisation of DNN and porting to GPU (OpenACC, OpenCL, CUDA)
    - Asynchronous parallel implementation of Stochastic Gradient Descent
    - Compression of DNN and porting them to GPU
  - Cluster parallelization using Spark
    - using PySpark (Python API to Spark)
  - Further integration of TMVA in Jupiter notebook
    - Javascript TMVA GUI, interactive training mode

Conclusions

- Many recent developments are happening in ROOT / TMVA
  - new features, new interfaces and various improvements
    - we are innovating TMVA with the community and under the scope the IML
    - strong growing development team
- Feedback on the new features is very welcomed !
- Easy to contribute
    - everybody interested is welcomed to join the development team
    - or can contribute via pull requests on ROOT github:

      https://github.com/root-mirror/root