Jupyter and TMVA

Attila Bagoly (Eötvös Loránd University, Hungary)

Mentors:
- **Sergei V. Gleyzer**
- **Enric Tejedor Saavedra**

Motivation

- Jupyter notebook:
  - Portable: you just need a browser
  - Interactive coding
  - Document: HTML, Markdown support
  - Shareable: SWAN, nbviewer, binder

- Integrating TMVA in Jupyter:
  - Support for TMVAGui in notebooks
  - Classifier visualizations
  - Pythonic interface for a bunch of functions
  - Interactivity: changes modify the state of TMVA
  - HTML formatted output

Code structure

- Importing ROOT will import JsMVA

- %jsmva on: JPyInterface inserts new methods to TMVA.DataLoader and TMVA.Factory

- New methods: inserting HTML to cell output, with JavaScript call

- Using: JsROOT to create plots
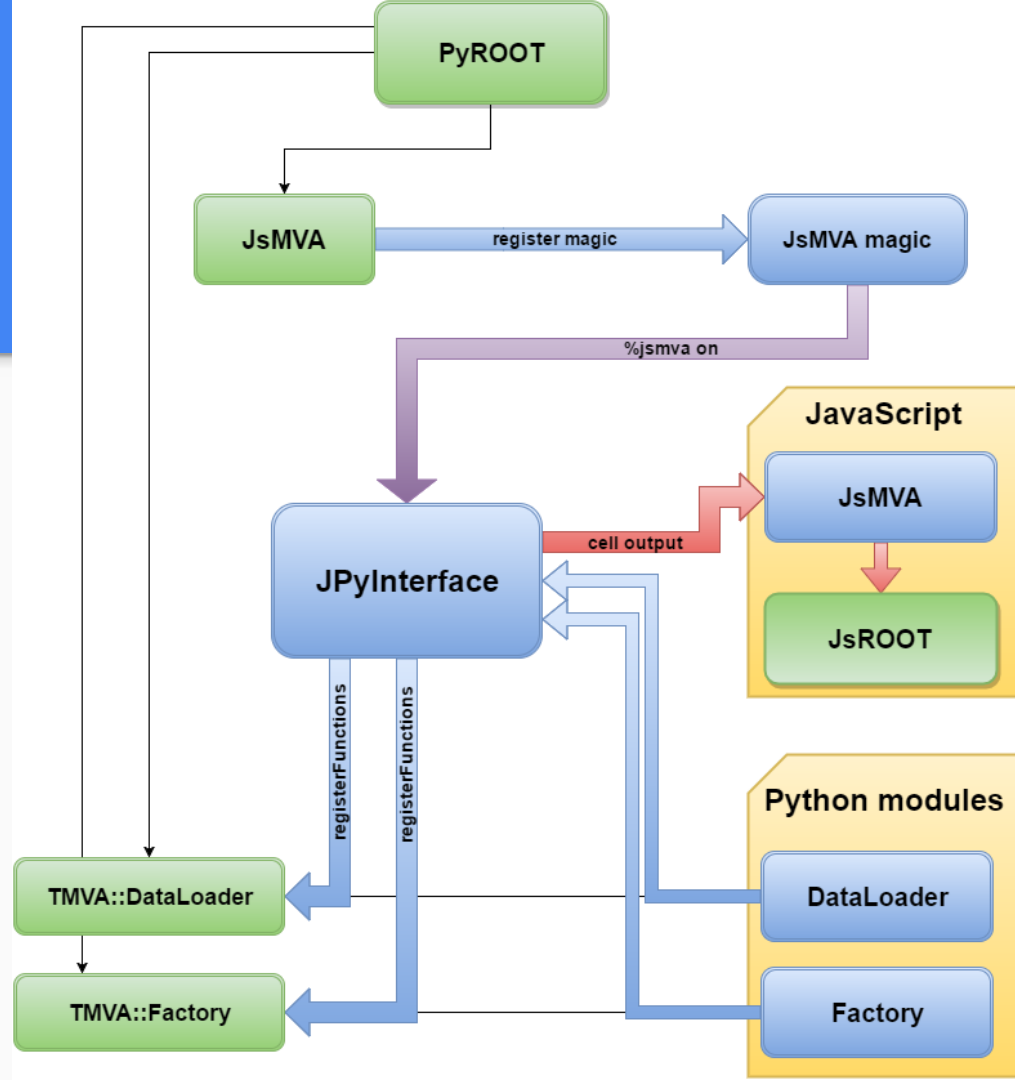
TMVAGui visualizations


```
loader.DrawCorrelationMatrix("Signal")
```

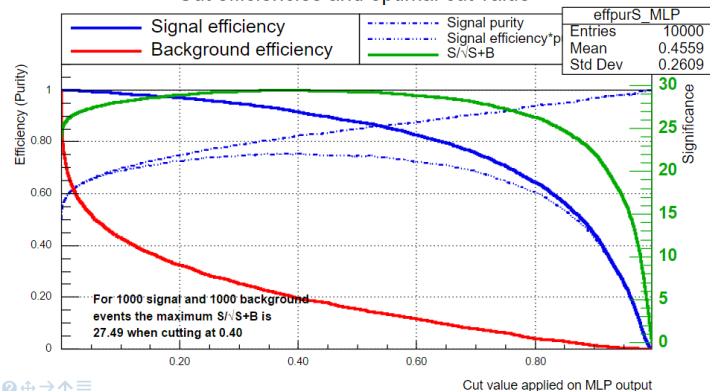**Visualizations related to input variables**

- Correlation matrix
- Input variables
- Transform input variables & show

Visualizations related to classifier outputs

- ROC curve
- Output distributions
- Cut efficiencies

```
factory.DrawCutEfficiencies(dataset, "MLP")
```

4

# Classifier output: Neural networks, decision trees

Simple neural network

- Python function reads the network, converts to JSON; JS with d3js make the visualization from JSON
- Interactive: focusing connections, zooming, moving
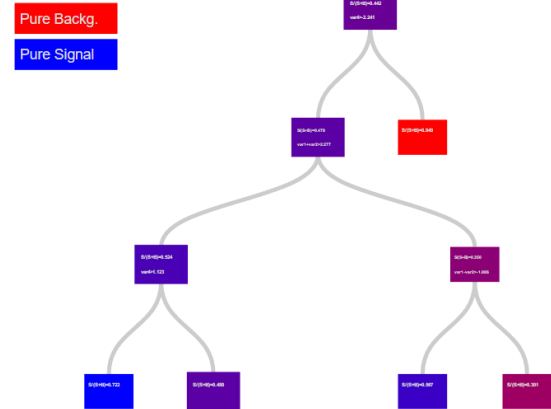
Deep neural network

- HTML5 Canvas visualization (speed)
- Less interactive: zooming, moving

Decision trees

- Ipywidgets: input field for selecting the tree
- Visualization from JSON with D3js
- Interactive: closing subtree, showing the path, focusing, moving, zooming, reset
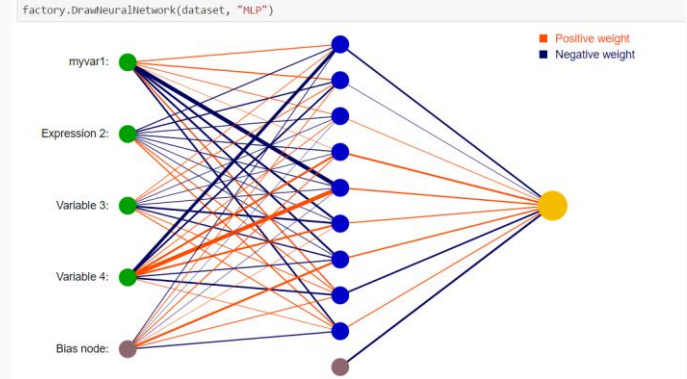
Interactive training mode

- C++ interface for tracking/stopping the training

- New thread for training

- Main thread periodically refreshes the plot (inserts small JS script, which removes itself)

- Error plots supported for MLP, DNN, BDT methods

- Progress bar for a bunch of methods

- Stop button: by clicking on it the main thread will send stop message for training loop (just the loop, no interfere with saving the net, or other data)

Train method: BDT

Stop

100%

■ Boost weight
■ Error Fraction

Train method: Cuts

Stop

54%
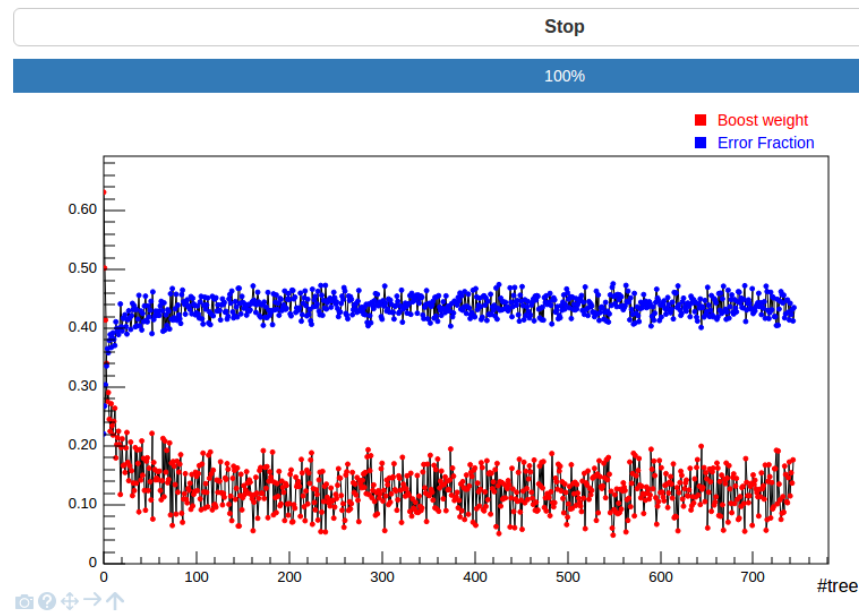
Python based user interface

```
factory = TMVA.Factory(JobName="TMVAClassification", TargetFile=outputFile,
                       V=False, Color=True, DrawProgressBar=True, Transformations=["I", "D", "P", "G", "D"],
                       AnalysisType="Classification")
```

New interface:

- Bunch of functions use option string

- Python: cleaner input without option strings

- JsMVA replaces these functions with a corresponding wrapper functions

- The settings can be passed by named arguments: V=True,Transformations=["I","D"] will be translated to "!V:Transformations=I,D"

Arguments of constructor: The options string can contain the following options:

Keyword	Can be used as positional argument	Default	Predefined values	Description
JobName	yes, 1.	not optional	-	Name of job
TargetFile	yes, 2.	if not passed histograms won't be saved	-	File to write control and performance histograms histograms
V	no	False	-	Verbose flag
Color	no	True	-	Flag for colored output
Transformations	no	""	-	List of transformations to test. For example with "I;D;P;U;G" string identity, decorrelation, PCA, uniform and Gaussian transformations will be applied
Silent	no	False	-	Batch mode: boolean silent flag inhibiting any output from TMVA after the creation of the factory class object
DrawProgressBar	no	True	-	Draw progress bar to display training, testing and evaluation schedule (default: True)
AnalysisType	no	Auto	Classification, Regression, Multiclass, Auto	Set the analysis type

Annotated example:
http://nbviewer.jupyter.org/github/qati/GSOC16/blob/master/notebooks/ROOTbooks-TMVA-JsMVA-UserInterface.ipynb

Deep neural network builder

- Booking DNN: lots of settings, everybody forgets the exact names

- Graphical interface: intuitive way to book DNN

- We can add different types of layers

- Specify the neuron number and training strategy for layer

- Connect the layers: building the network

- Save network: transform the graphical representation to option string and books the method

HTML formatted output

- Output from TMVA methods are transformed into HTML

- CSS for style description

- Cleaner and interactive

	Number of training and testing events		
Signal	training events	2400	
	testing events	600	
	training and testing events	3000	
Background	training events	2400	
	testing events	600	
	training and testing events	3000	
DataSetInfo	Correlation matrix (Signal)		
DataSetInfo	Correlation matrix (Background)		
DataSetFactory	**Dataset: BDT_fold4**		
Factory	Train method: BDT for Classification		
BDT	#events: (reweighted) sig: 2400 bkg: 2400		
	#events: (unweighted) sig: 2400 bkg: 2400		
	Training 2 Decision Trees ... patience please		
	Elapsed time for training with 4800 events : 0.0133 sec		
BDT	**Dataset: BDT_fold4** Evaluation of BDT on training sample (4800 events)		
	Elapsed time for evaluation of 4800 events : 0.00358 sec		
	Creating xml weight file: BDT_fold4/weights/TMVAClassification_BDT.weights.xml		
	Creating standalone class: BDT_fold4/weights/TMVAClassification_BDT.class.C		

Future of JsMVA

- Visualizations for other classifiers

- Improve interactive training (open closed notebook and continue training, new interactive plots)

- Visualization for regression, multiclass classification, autoencoders

- We are open to new suggestions

10

Summary

- JsMVA can be activated easily, by using %jsmva on command

- Visualizer methods inserted to TMVA::Factory, and TMVA::DataLoader

- Implemented TMVAGui visualizations

- Classifier visualizations: MLP, DNN, BDT

- Interactive training mode: tracking the progress and the erros

- New input mode (no more option strings), new output (HTML formatted)

- DNN builder: graphical interface for booking DNN

List of new features

- TMVAGui:
    - Input variable and transformed input visualization
    - Correlation matrix
    - Classifier output/probability distributions
    - Classifier cut efficiencies

- Classifier visualizations:
    - Simple neural network
    - Deep neural network
    - Decision Tree

- Interactive training mode

- New user interface

- HTML formatted output

- Deep neural network builder

12

Coming soon to SWAN gallery with the next ROOT release

Everything on GitHub:

https://github.com/qati/GSOC16

Notebooks on nbviewer (static, rendered):

http://nbviewer.jupyter.org/github/qati/GSOC16/blob/master/index.ipynb

Notebooks on binder (interactive):

www.mybinder.org/repo/qati/GSOC16

Thank you for your attention!