Status of Multiclass in TMVA

Kim Albertsson (CERN/LTU)
2017-03-17

Outline

- Overview of what multiclass is already in TMVA
- Motivation for this work
- Visualisation of multiclass performance
- Fixes to TMVA multiclass

Overview

- Has been in since 2010, but not advertised
- Original work by Jan Therhaag and Joerg Stelzer
- Multiclass example + gui
    - with bdt, mlp pde-foam, fdga, and (surprisingly dnn)

Motivation

- People working with ttH analysis Stefan Gadatsch, Andreas Hoecker et al.

- Simultaneous fit of signal and several distinct background channels to decrease uncertainties

- They use external tools since status of TMVA multiclass is unknown

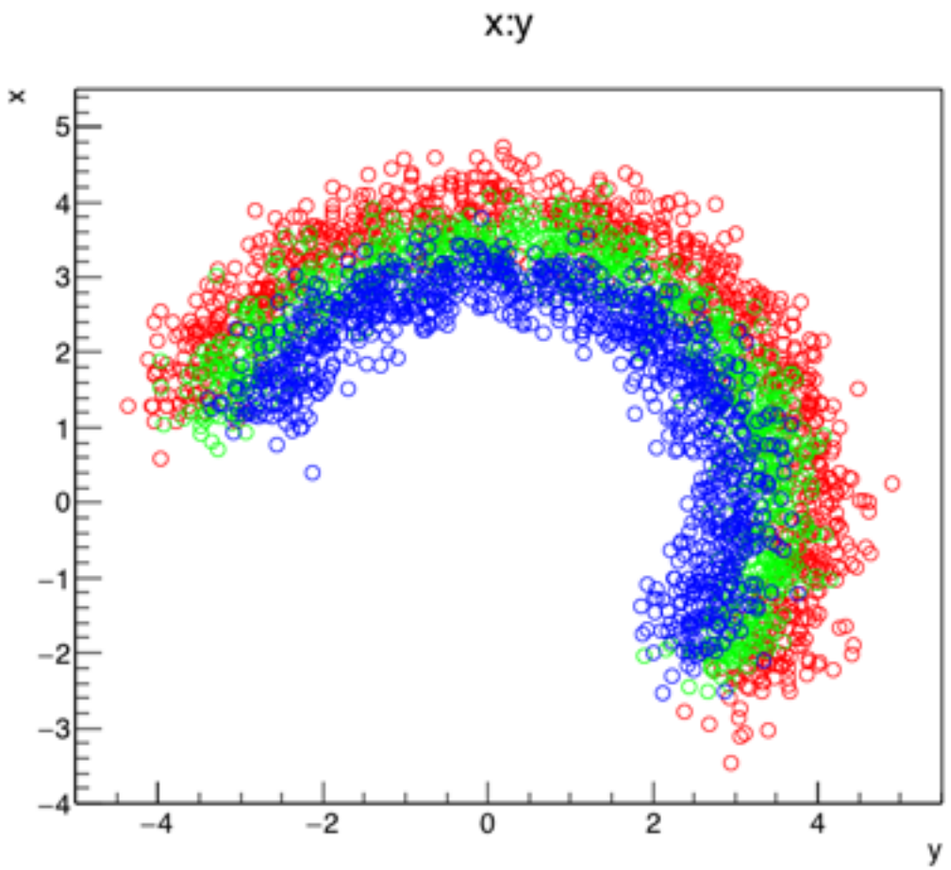
- I provide performance benchmarks between TMVA and the external tools

Visualisation

- Toy example
- mvaweights.C
- Multiclass ROC
- Performance matrix

Visualisation - Toy example

- Three classes
- Non-linear decision boundary
- Especially tricky to pick out the middle class.

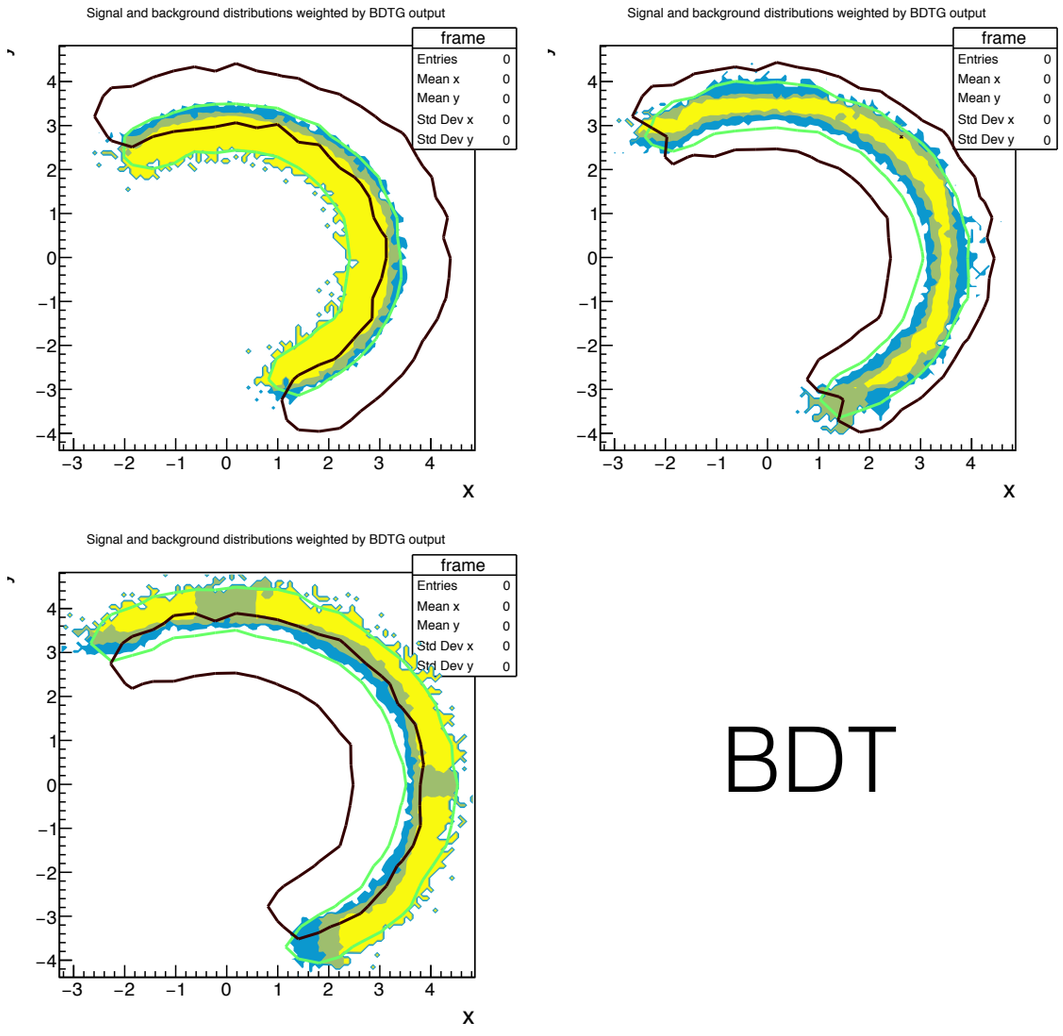


Visualisation - Multiclass mvaweights

- Shows where output of classifier is strong (yellow) and weak (white)

- Shows characteristics of different classifiers

- Gradient boosted decision trees adapt well



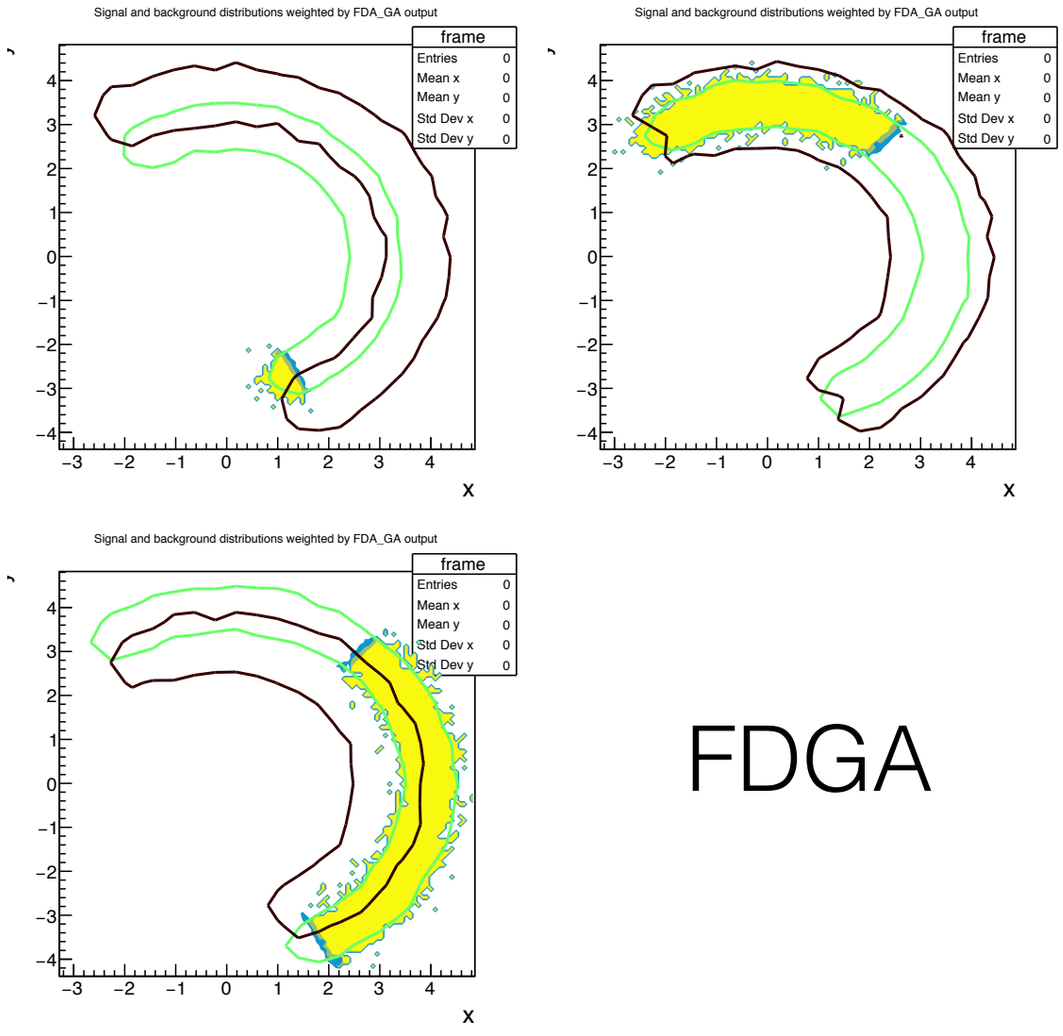
BDT

Visualisation - Multiclass mvaweights

- Shows where output of classifier is strong (yellow) and weak (white)

- Shows characteristics of different classifiers

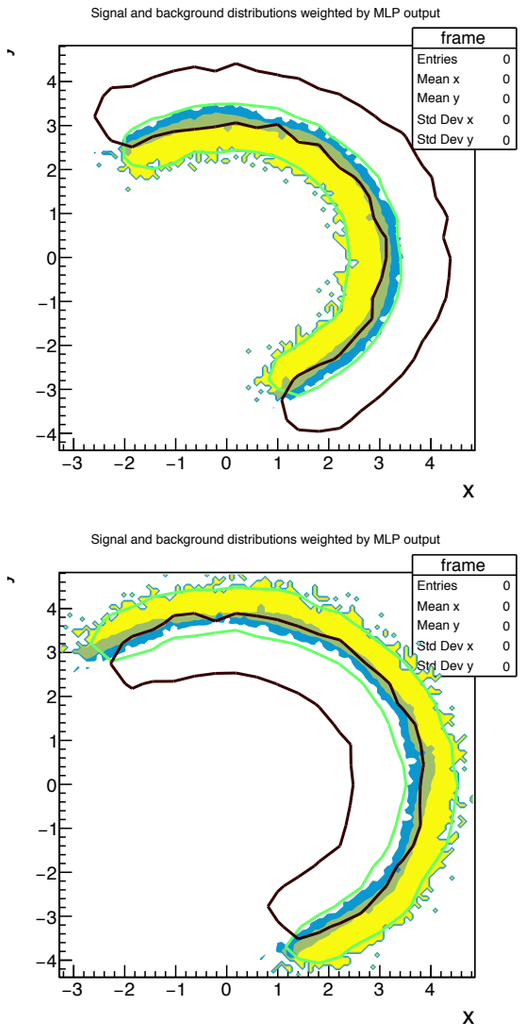
- Fischer linear discriminant, not as well

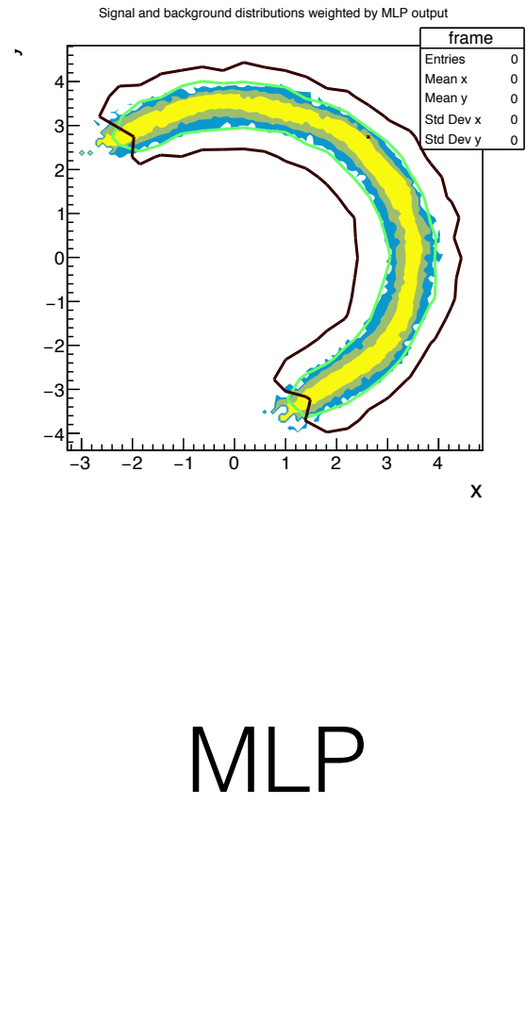


FDGA

Visualisation - Multiclass mvaweights

- Hardest part was making the macro understand the TTree structure
- Will be included in the macro folder

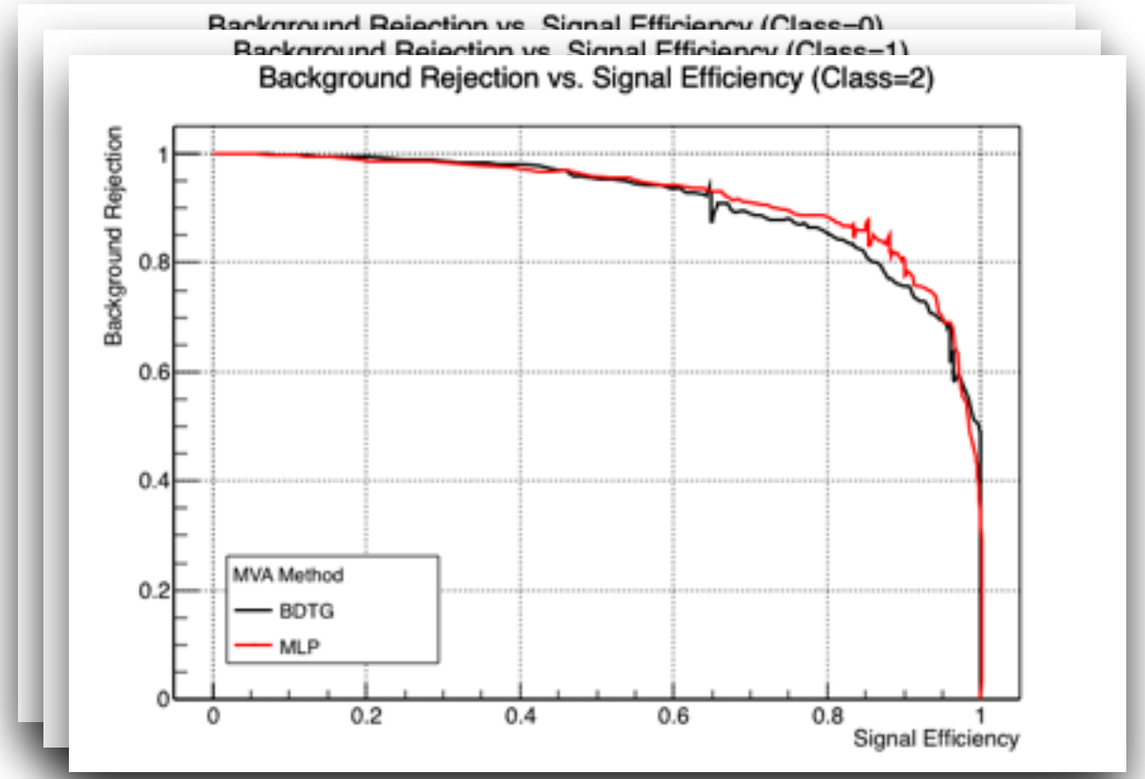




MLP

Visualisation - Multiclass ROC

- Simple extension ROC to multiclass (1 vs rest)
- One graph per class
- Todo class labels instead of numbers

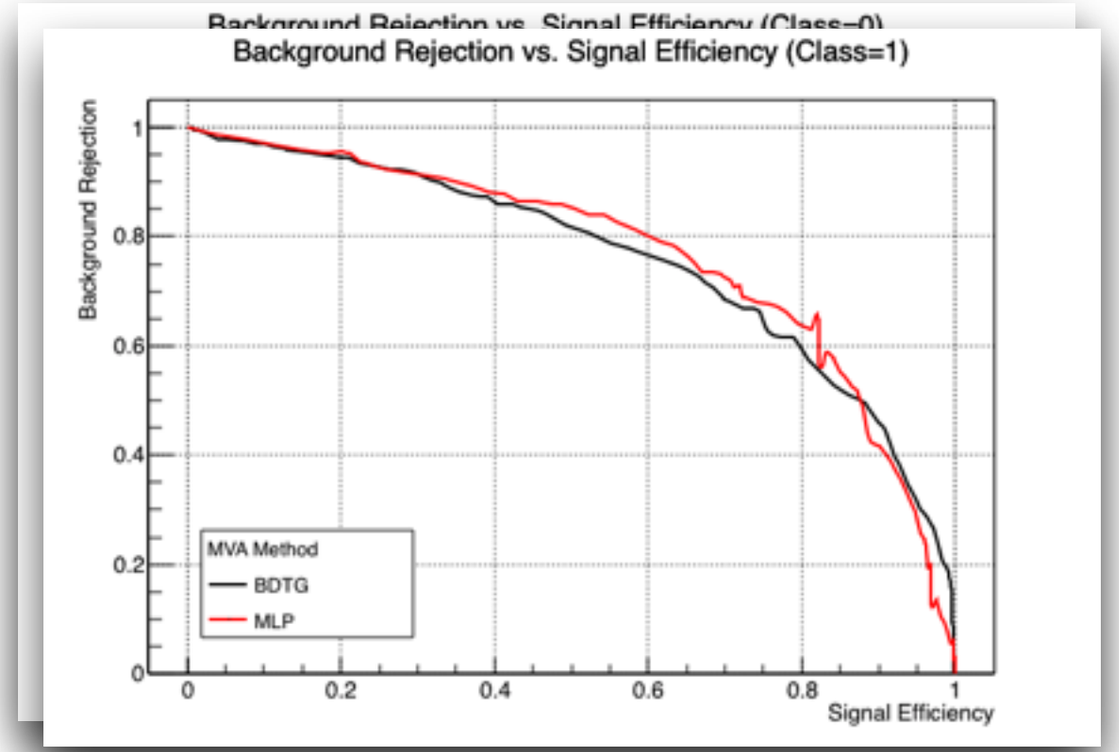


Visualisation - Multiclass ROC

- First step integrated into TMVA, open PR 380 (just merged)

- Integration with TMVAMulticlassGUI ongoing

- When is next release?



Visualisation - Performance Matrix

- Compare all classes against all other classes N*N (1 vs 1)
  - ROC integral
    - Efficiency/Purity at working point
- Textual and graphical output planned
- Only textual for ROC integral implemented yet

```
Line: 176 - Processing method BDTG
        A         B         C
    A   1.0000    0.8175    0.9790
    B   0.8799    1.0000    0.8299
    C   0.9804    0.6819    1.0000
Line: 176 - Processing method MLP
        A         B         C
    A   1.0000    0.6192    0.9842
    B   0.8829    1.0000    0.8561
    C   0.9872    0.6992    1.0000
```

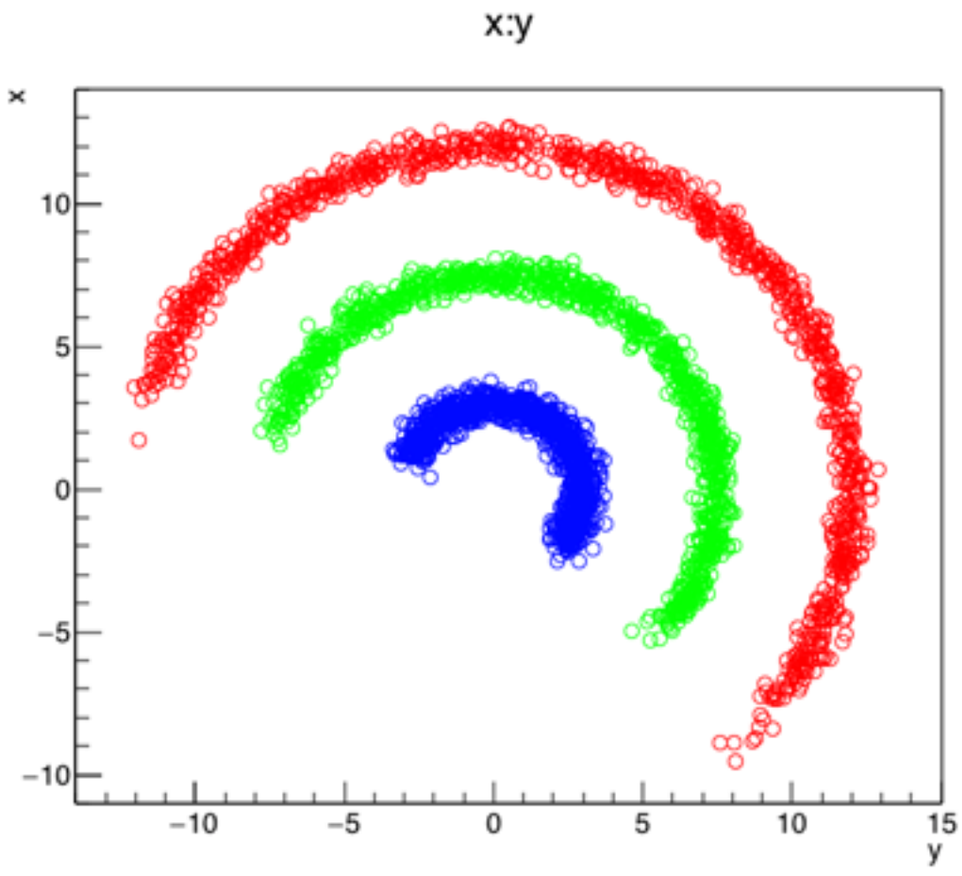
Fixes

- MLP output

- Multiclass `Factory::EvaluateAllMethods`

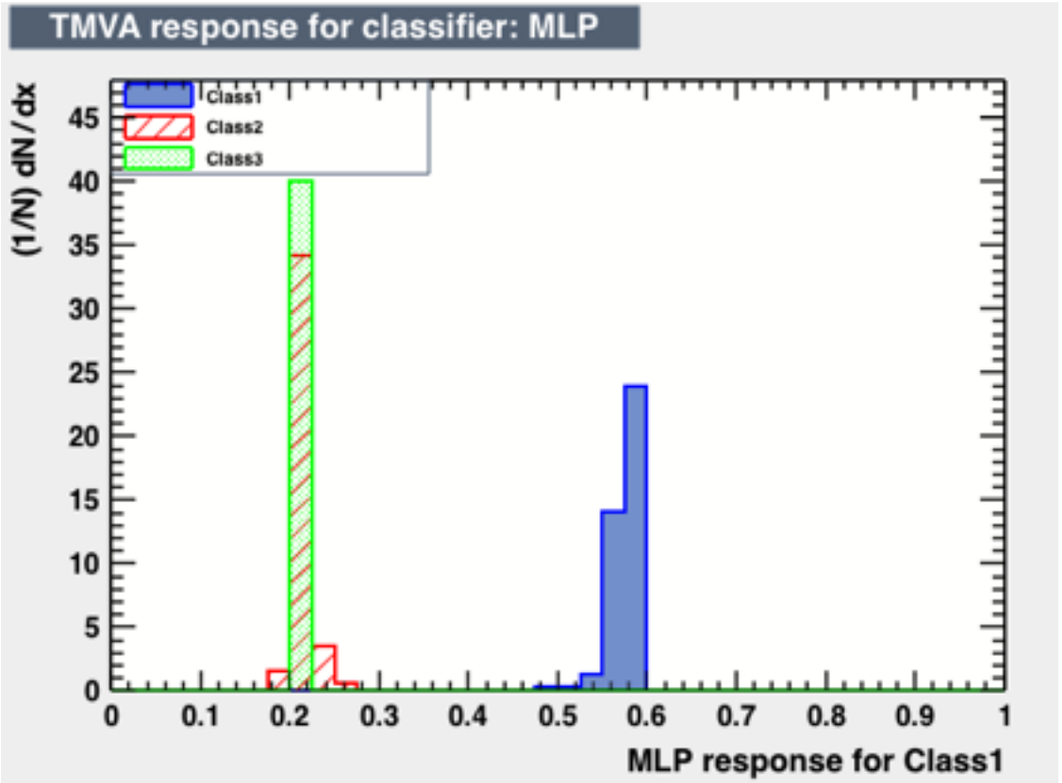
Fixes - Toy example

- Three classes
- Completely separated



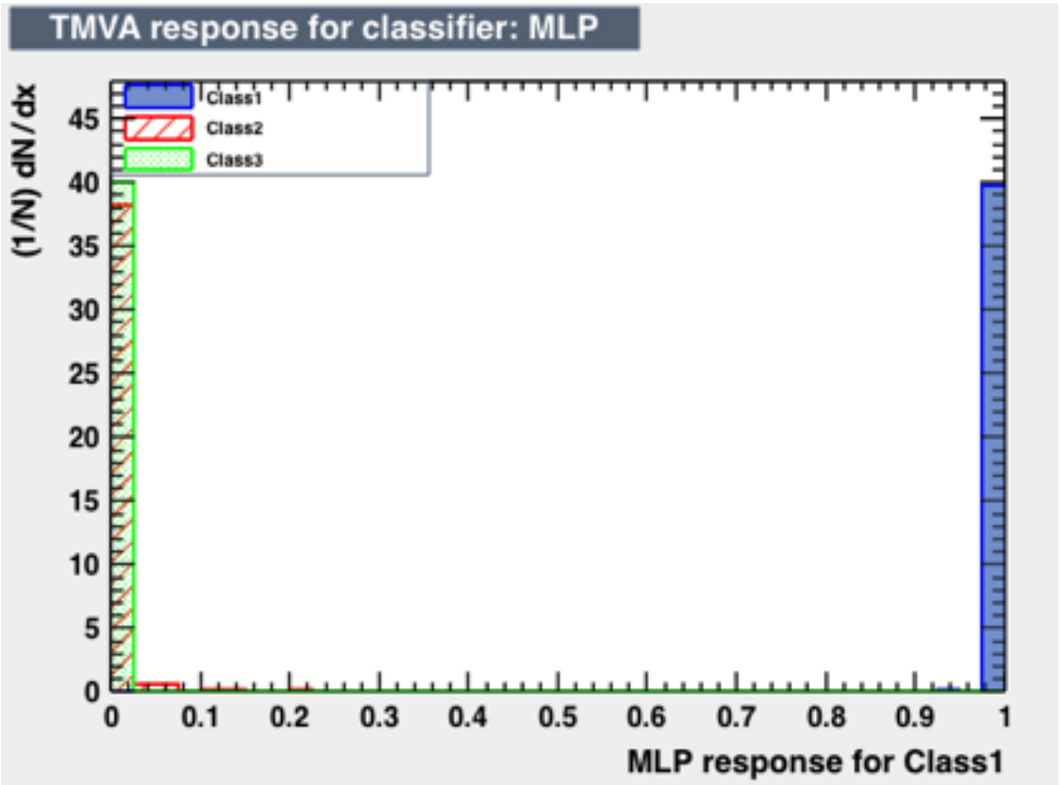
Fixes - MLP output

- MLP had narrow range even on completely separated data
- A softmax layer was missing for training but included for evaluation
- PR 422 for this fix, ready for integration
- We might want to have a second look at this



Fixes - MLP output

- MLP had narrow range even on completely separated data
- A softmax layer was missing for training but included for evaluation
- PR 422 for this fix, ready for integration
- We might want to have a second look at this



Fixes - Evaluation

- `Factory::EvaluateAllMethods` was taking as long as training for my particular case (10 min)

- Problem was inefficient reading of data

- Runtime reduced an order of magnitude, at the cost of increased temporary memory usage

Fixes - Evaluation

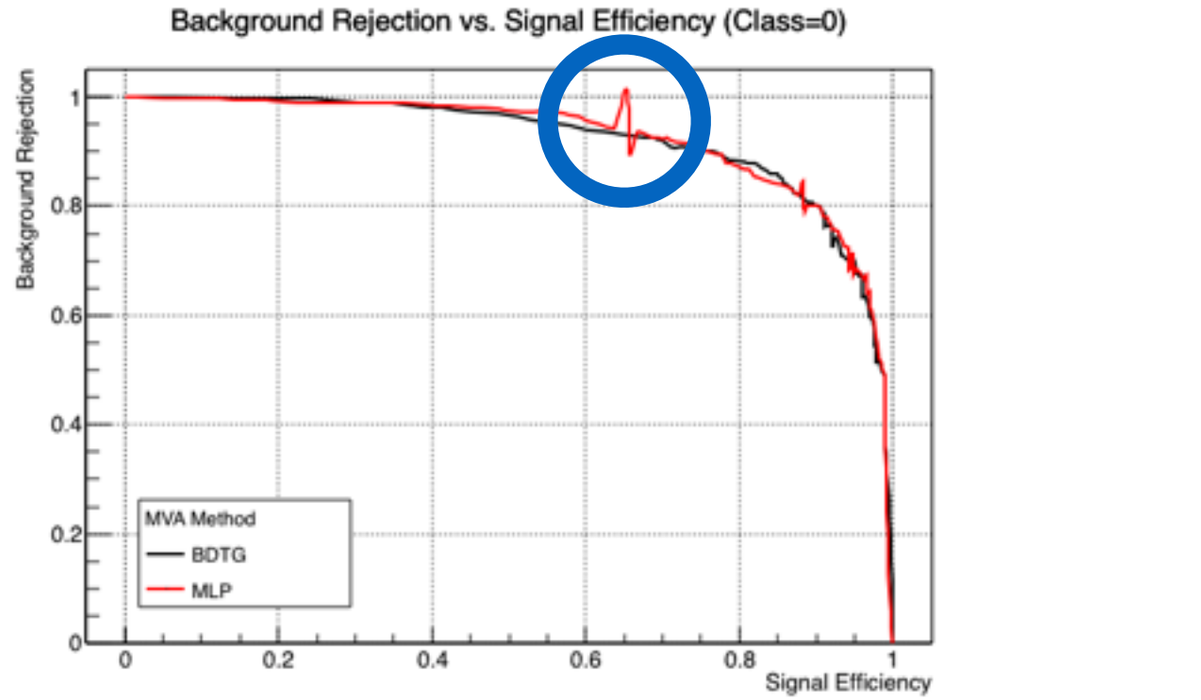
- No PR for this yet since this was just done for fun

- ROOT team is requesting this fix (PR 438), the time consumption is making tests time out

Thanks

Extras - ROC curve plotting

- The ROC curve should be monotonically decreasing
- Problem now with double points and the chosen interpolation



Extras - ROC curve plotting

- Use simpler interpolation
    - Strong suggestion, but does not solve all problems
- Uniform sampling of target domain instead of classifier output domain
    - Better and more reliable output