# Introduction to Boosted Decision Trees

# A multivariate approach to classification problems

**Christian Böser, Simon Fink, Steffen Röcker**
**Institut für Experimentelle Kernphysik, KIT**

# Overview

- Introduction
- Decision trees
  - Learning, Gini Index
  - Random Forests
  - Boosting
  - ROC curve, Overtraining, Examples, ...
- Demonstration of Boosted Decision Trees in TMVA

All tools are freely available, see

ROOT+TMVA: root.cern.ch, tmva.sf.net

Used macros: http://goo.gl/UY4QSa

# Introduction

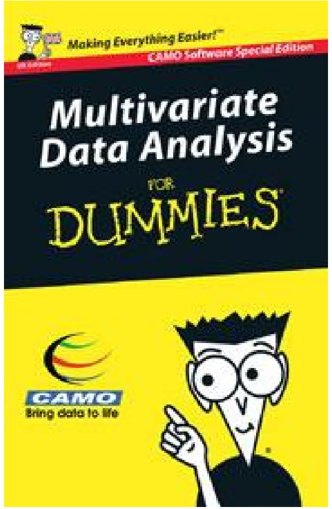
- An often faced problem is to **predict the answer** to a question based on different input variables
- Two different problems:
  - **Classification**
    - Predict only a binary response
      - Do I need an umbrella today? → *Yes/No*
      - What is the measured data? → *Signal/Background*
  - **Regression**
    - Predict an exact value as an answer
      - What will be the temperature tomorrow? → *-19 °C, 7 °C, 38 °C, …*
      - What will be your life expectancy? → *35 yrs, 78 yrs, 122 yrs, …*
- This session will only cover the **classification** problem

# Why Multivariate?

- Allows to combine several discriminating variables into one final discriminator $R^d \rightarrow R$
- Better separation than one variable alone
- Correlations become visible

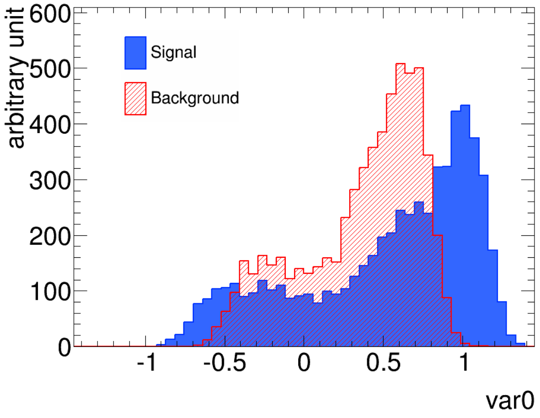
Available methods:

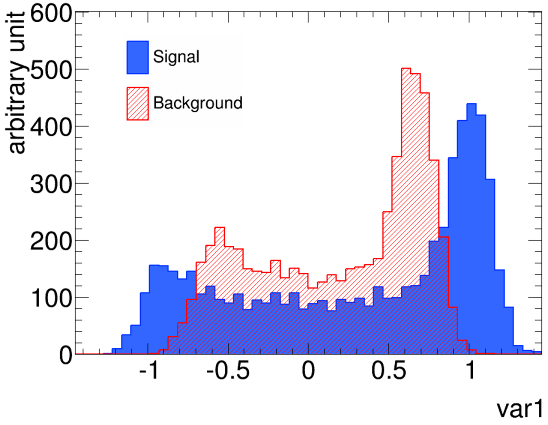
- **Boosted Decision Trees**
- Neural Networks
- Likelihood Functions
- ...



# 2-dimensional Example

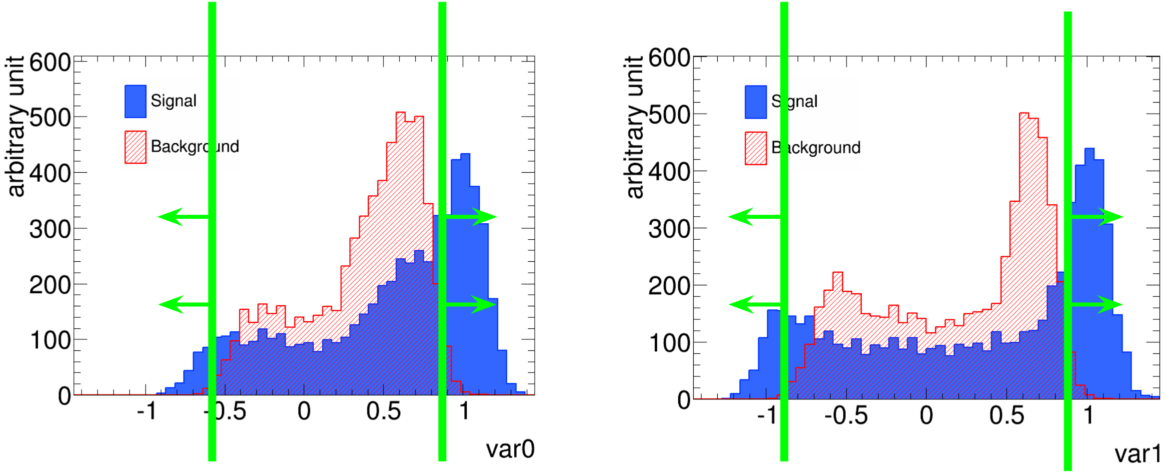
- Why use a multivariate analysis? Why not simply cut on variables?
  - Correlations not visible



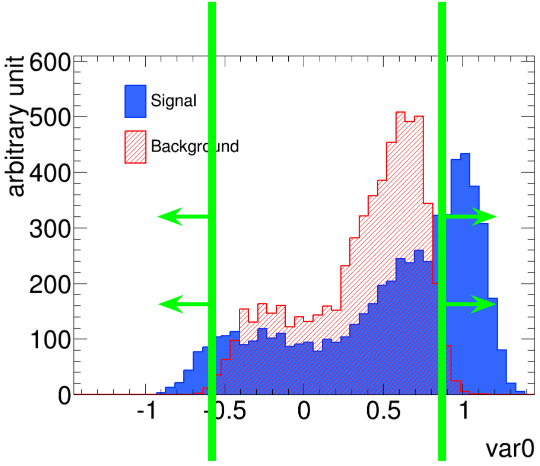
# 2-dimensional Example

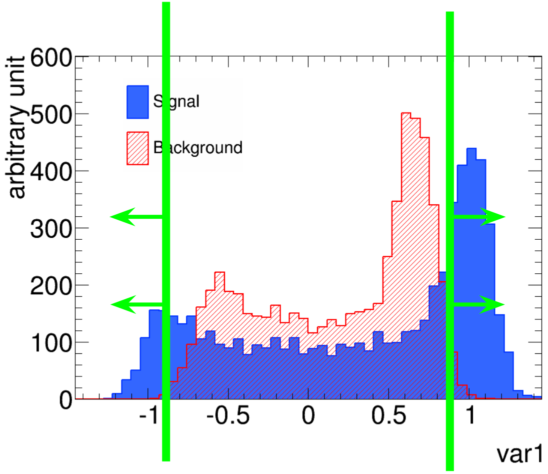
- Why use a multivariate analysis? Why not simply cut on variables?
  - Correlations not visible

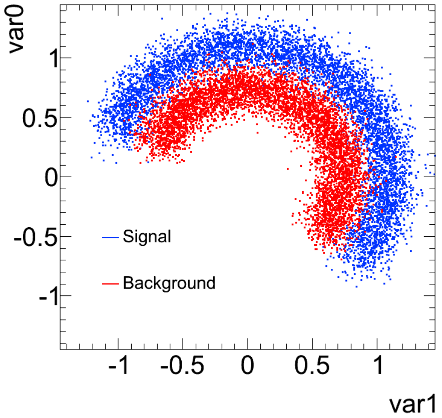


# 2-dimensional Example

- Why use a multivariate analysis? Why not simply cut on variables?
  - Correlations not visible

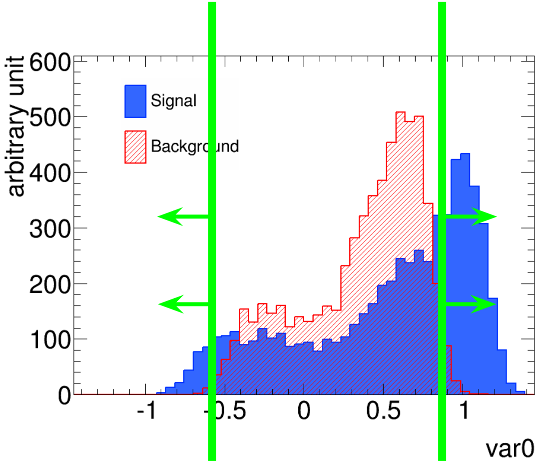


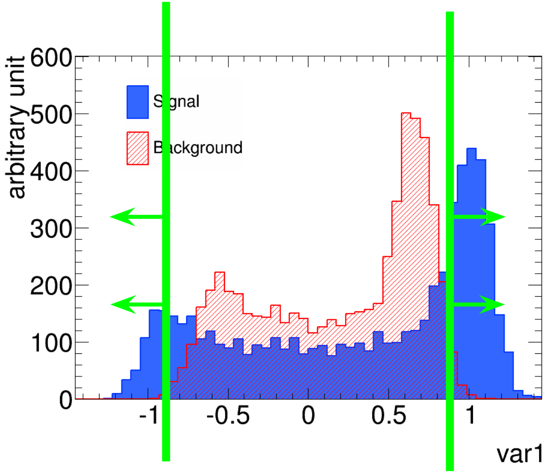



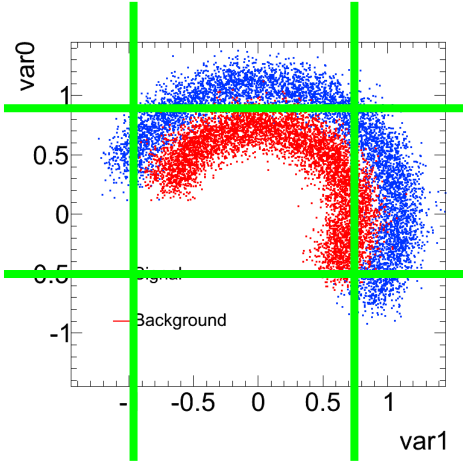



# 2-dimensional Example

- Why use a multivariate analysis? Why not simply cut on variables?
  - Correlations not visible

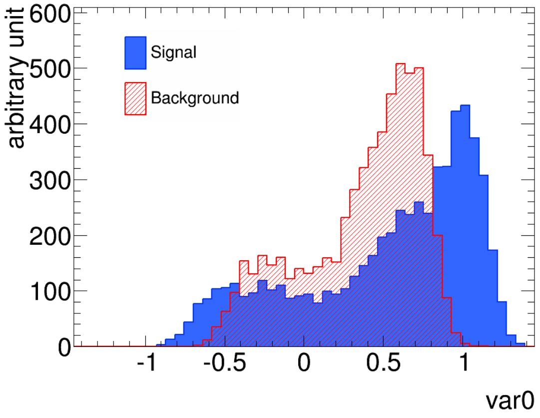


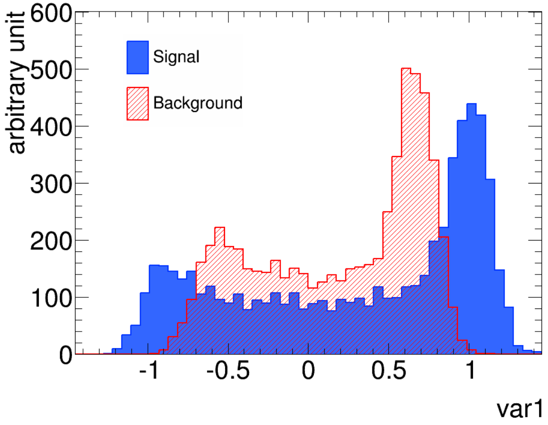



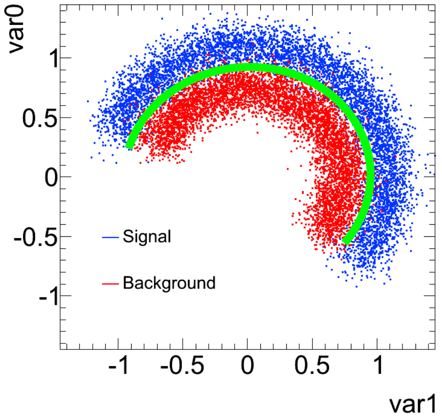



# 2-dimensional Example

- Why use a multivariate analysis? Why not simply cut on variables?
  - Correlations not visible
  - The more dimensions, the better the possible separation power

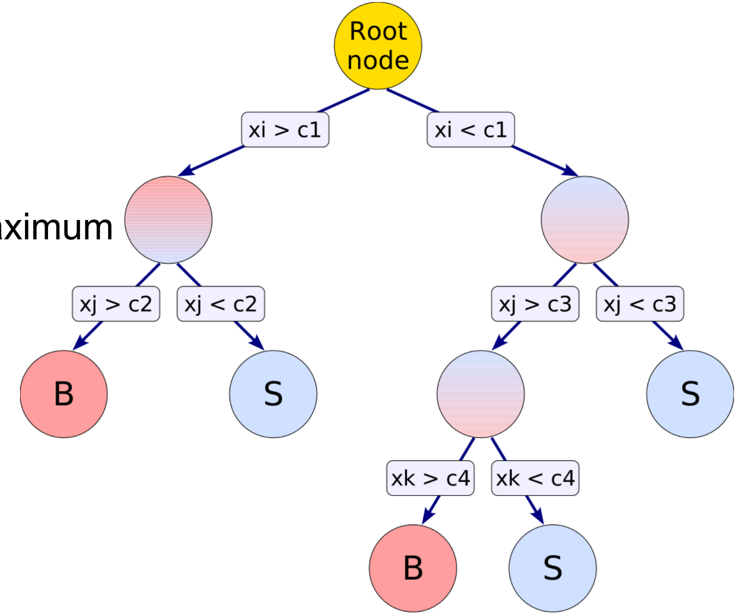


# Decision Tree

- What exactly is a Decision Tree?
  - Consecutive set of questions (**nodes**)
  - Only two possible answers per question
  - Each question depends on the formerly given answers
  - Final verdict (**leaf**) is reached after a given maximum number of nodes




- Advantages of Decision Trees
  - Easy to understand/interpret
  - Good with multivariate data
  - Fast training
- Disadvantages
  - Single tree not very strong → Random Forests

# Decision Tree Learning

- But how to create a DT?
  - A DT needs to be *trained* on a dataset which already provides the outcome (e.g Simulation dataset with signal and background processes)
  - Choice of node criterion by maximizing **separation gain** between nodes

$$\text{separation gain} \cong \text{gain}(\text{parent cell}) - \text{gain}(\text{daughter cell 1}) - \text{gain}(\text{daughter cell 2})$$

  - gain can be computed in different ways, a common used one is **Gini Index**

$$gain(cell) \cong p \cdot (1-p), \quad p : \text{Purity}$$

- Repeat until the maximum number of nodes is reached

# Random Forests

- Random Forests is an ensemble method that combines different trees
- Final output is determined by the majority vote of all trees

A Random Forest combines the votes of all trees

# Random Forests

- Random Forests is an ensemble method that combines different trees
- Final output is determined by the majority vote of all trees
- The idea is, that a sum of weak learners results in a stronger learner

Simple example:

- 3 different trees which are uncorrelated and are correct in 60% of cases
- In order to correctly classify an event, only ⅔ trees have to be correct. That means, the misclassification probability is either 3 wrong or ⅔ wrong:
  - $P = \binom{3}{2} * 0.4^2 * 0.6 + \binom{3}{3} * 0.4^3 * 0.6^0 = 0.352$
- Therefore the ensemble of trees is better than only one tree even though their separation power is the same (if uncorrelated)

# Random Forests

- Random Forests is an ensemble method that combines different trees
- Final output is determined by the majority vote of all trees
- The idea is, that a sum of weak learners results in a stronger learner

Available methods to train Random Forests:

- Bagging
  - A subset of events is drawn from the training data with replacement
  - Tree is trained on this subset, this is repeated many times
- Boosting
  - Misclassified events are weighted higher so that future learners concentrate on these
  - Most commonly used method → AdaBoost

# Boosting

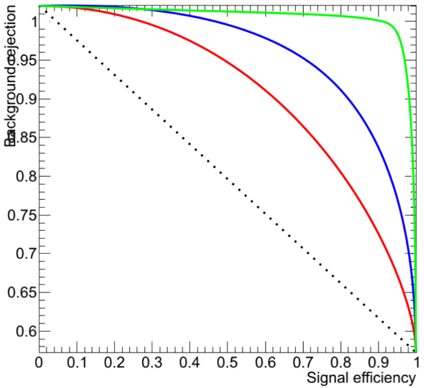
- AdaBoost (Adaptive Boosting):
  - Enhances weights of misclassified events and reduces weight of correctly classified ones after each training so that future trees learn those better
  - Iterates until weight of misclassified > 50%
  - Final weight is the sum of all classifiers weighted by their errors

**w = (1-err)/err**

# ROC Curves

- ROC (**R**eceiver **O**perating **C**haracteristic) Curves are a good way to illustrate the performance of given classifier
- Shows the background rejection over the signal efficiency of the remaining sample
- Best classifier can be identified by the largest AUC (Area under curve)

- Neyman-Pearson lemma: The best ROC curve is given by the likelihood ratio L(x|S) /L(x|B)

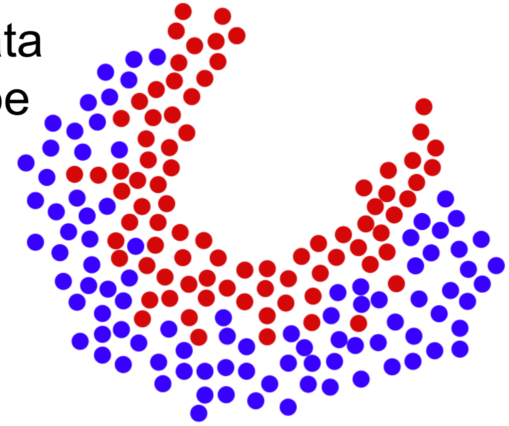


# Overtraining

- Boosted Decision Trees are very easy to overtrain, that is they will learn statistical fluctuations by heart

How to avoid it:

- One should split available data in training / test data
- Performance on the training samples should not be better than on the test sample
- Pruning can cut away insignificant nodes

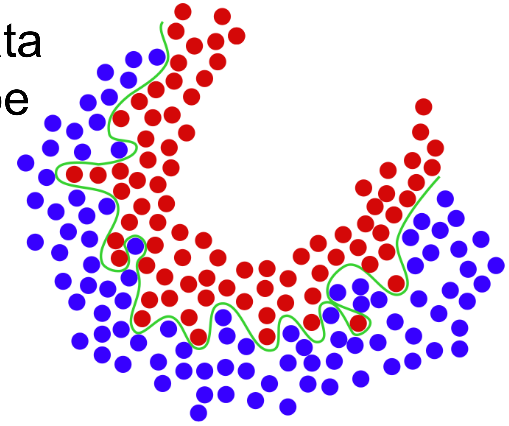


# Overtraining

- Boosted Decision Trees are very easy to overtrain, that is they will learn statistical fluctuations by heart

How to avoid it:

- One should split available data in training / test data
- Performance on the training samples should not be better than on the test sample
- Pruning can cut away insignificant nodes

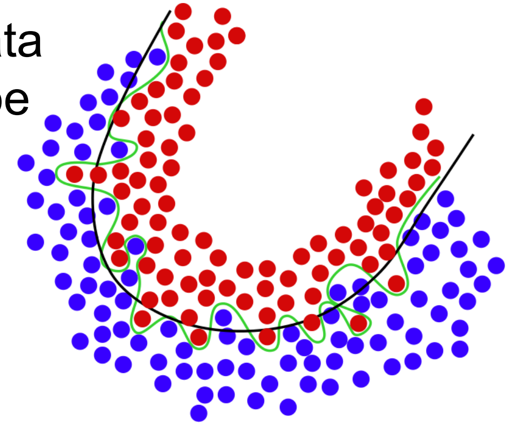


# Overtraining

- Boosted Decision Trees are very easy to overtrain, that is they will learn statistical fluctuations by heart

How to avoid it:

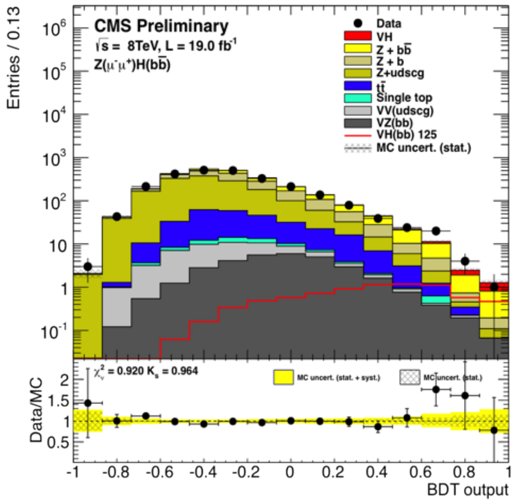
- One should split available data in training / test data
- Performance on the training samples should not be better than on the test sample
- Pruning can cut away insignificant nodes



# Real life examples

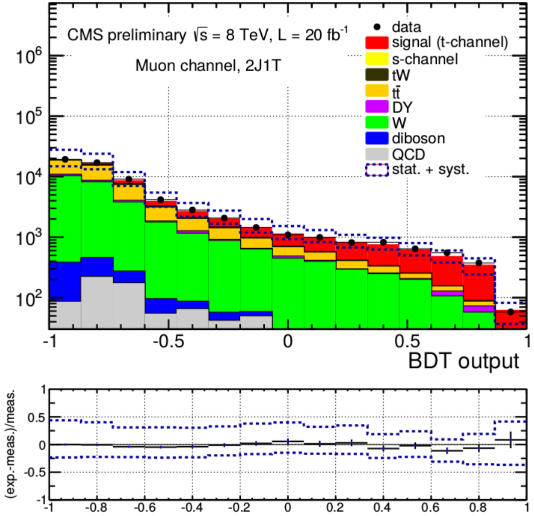
## Search for Higgs $Z(\mu^+\mu^-)H(b\bar{b})$

- Very few signal events
- Only “visible” by using MVAs




## Single Top Polarization

- Cut on BDT yields signal enriched sample
- Allows to study top quark properties



# Demonstration of TMVA

# TMVA

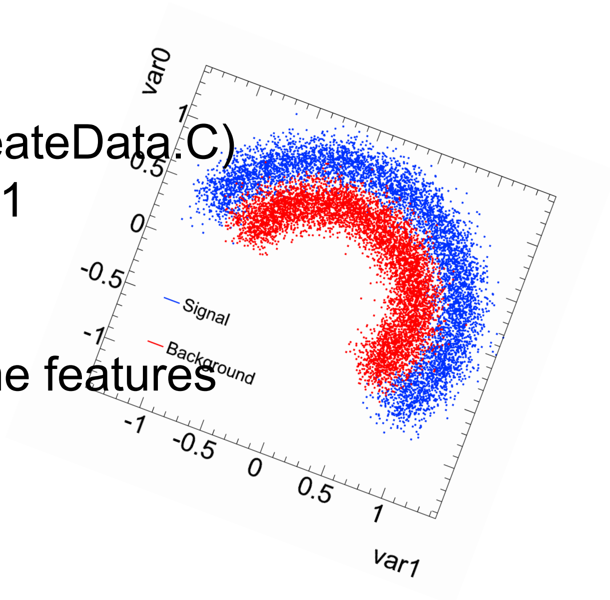
**T**oolkit for **M**ulti**V**ariate **A**nalysis:

- Part of ROOT
- Freely available
- Open source
- Many algorithms available: BDT, NN, LF, ...
- Commonly used in HEP

# Example with TMVA (Python)

Find the scripts in this dropbox: [http://goo.gl/UY4QSa](http://goo.gl/UY4QSa)

- Use same circular example from above
- data.root (created with $ROOTSYS/tmva/test/createData.C)
  - contains TreeS and TreeB with var0 and var1
- simpleplot.py
  - creates plots from slide 5
- trainBDT.py with 3 different settings to show some features
  - NTrees = 5
  - NTrees = 850
  - NTrees = 2500



# Example with TMVA (Python) II

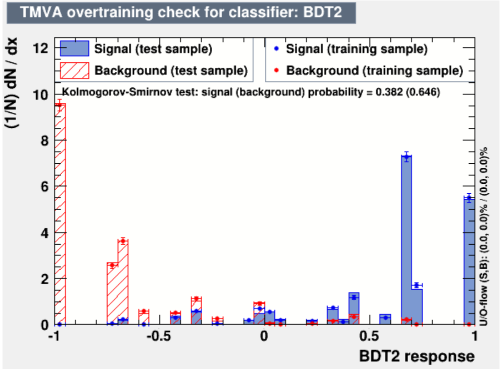
- Automatically split into training and testing samples (if no option given)
- Created files by trainBDT.py:
  - weight file in weights/ for application
  - output file test.root with all trainings and testing outcomes
- PlotDecisionBoundary.py:
  - use original data.root as input
  - applies training weights
  - draw decision boundaries
- PlotROC.py
  - use test.root as input
  - plot ROC curves of different trainings

# TMVA GUI (I)

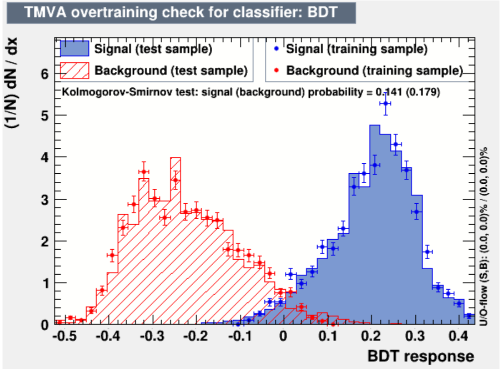
- Nice tool to produce lots of not-so-nice plots:
  - $ROOTSYS/test/tmva/TMVAGui.C'("test.root")'

- Option 4b (often used to check for overtraining):

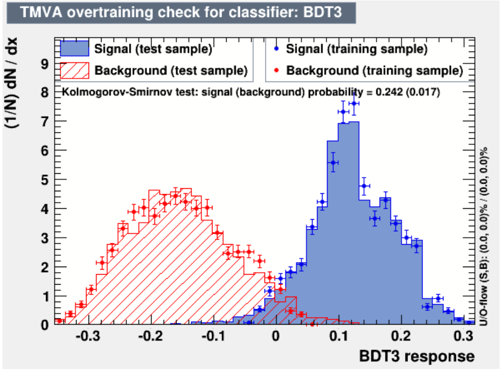
5 trees




850 trees

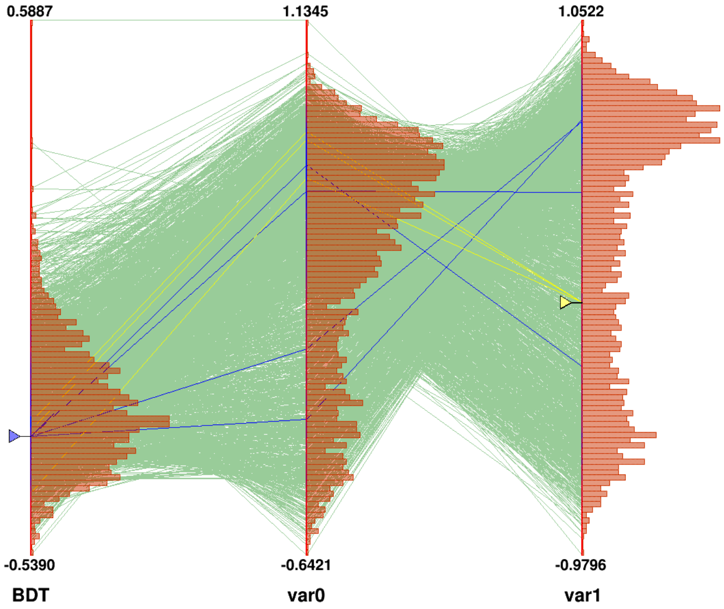



2500 trees



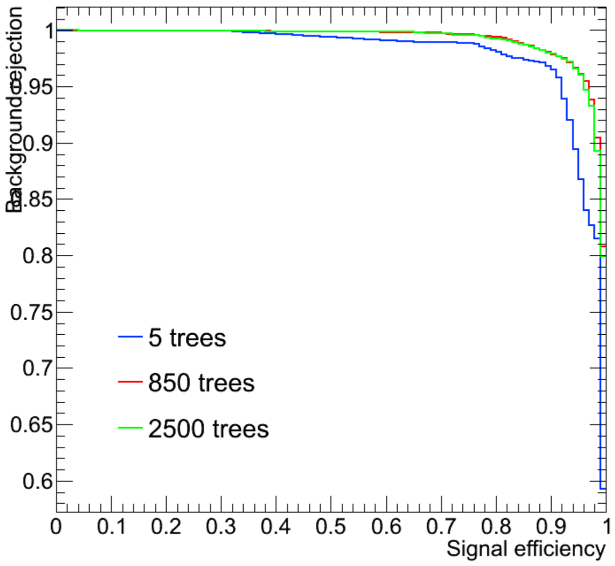
# TMVA GUI (II)

- Option 6 shows path of each event



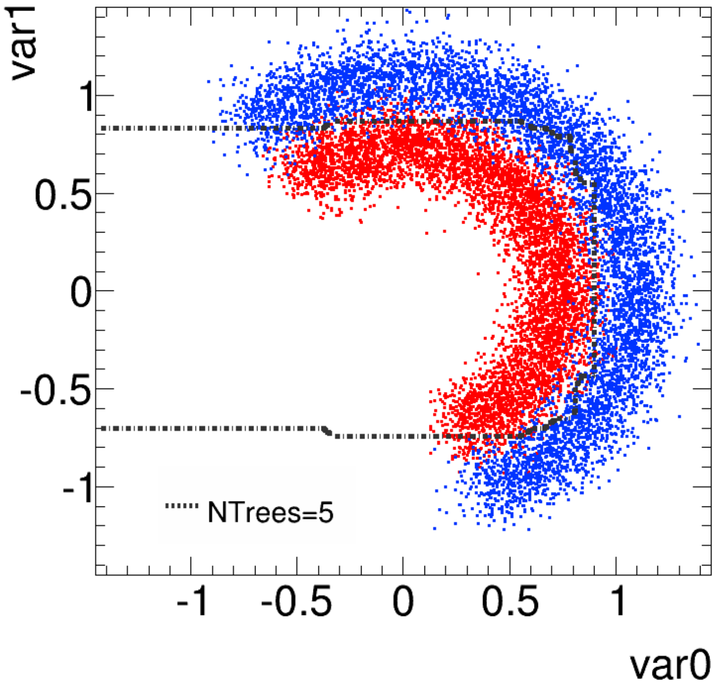
# ROC curves

- Check performance of MVA by plotting background rejection over signal efficiency
- Rule of thumb:
  - Use the training with the largest integral of ROC
- In this example:
  - Performance of 5 trees sub-optimal
  - 850 vs. 2500 trees almost identical



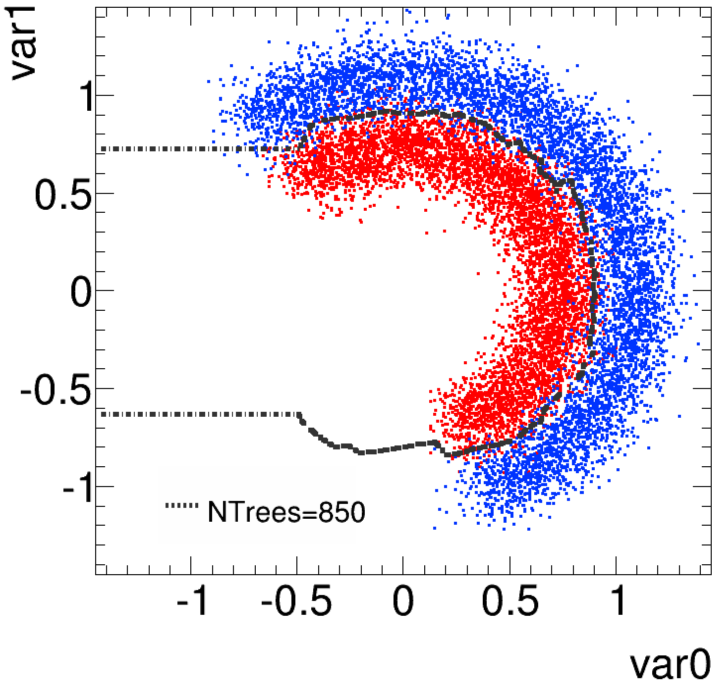
# BDT with 5 trees

- Not enough trees
- Sub-optimal separation between signal and background



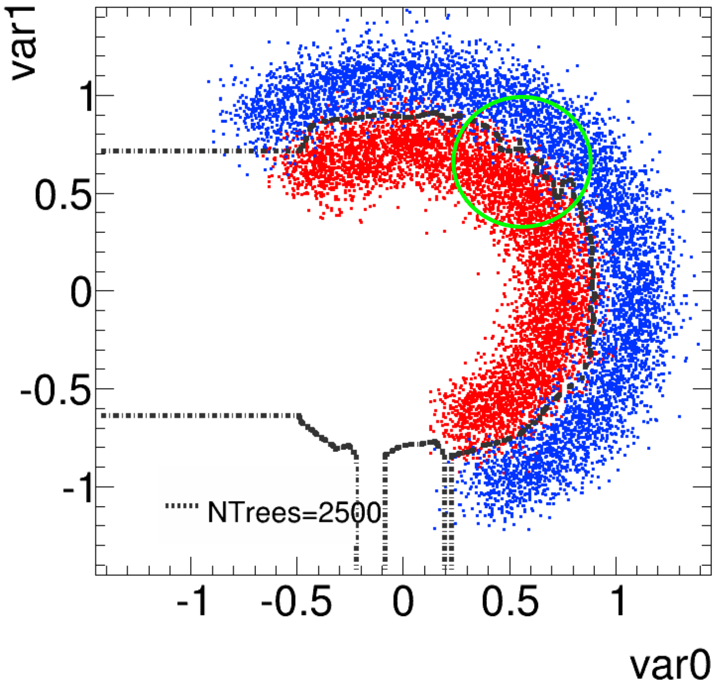
# BDT with 850 trees

- Close to best performance



# BDT with 2500 trees

- MVA learns fluctuations
- Tendency towards overtraining



# BACKUP

# From TMVA manual

| Criteria | | Cuts | Likelihood | PDE-RS / k-NN | PDE-Foam | H-Matrix | Fisher / LD | MLP | BDT | RuleFit | SVM |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MVA Method | | | | | | | | | |
| Performance | No or linear correlations | $\star$ | $\star\star$ | $\star$ | $\star$ | $\star$ | $\star\star$ | $\star\star$ | $\star$ | $\star\star$ | $\star$ |
| | Nonlinear correlations | $\circ$ | $\circ$ | $\star\star$ | $\star\star$ | $\circ$ | $\circ$ | $\star\star$ | $\star\star$ | $\star\star$ | $\star\star$ |
| Speed | Training | $\circ$ | $\star\star$ | $\star\star$ | $\star\star$ | $\star\star$ | $\star\star$ | $\star$ | $\circ$ | $\star$ | $\circ$ |
| | Response | $\star\star$ | $\star\star$ | $\circ$ | $\star$ | $\star\star$ | $\star\star$ | $\star\star$ | $\star$ | $\star\star$ | $\star$ |
| Robustness | Overtraining | $\star\star$ | $\star$ | $\star$ | $\star$ | $\star\star$ | $\star\star$ | $\star$ | $\circ$ | $\star$ | $\star\star$ |
| | Weak variables | $\star\star$ | $\star$ | $\circ$ | $\circ$ | $\star\star$ | $\star\star$ | $\star$ | $\star\star$ | $\star$ | $\star$ |
| Curse of dimensionality | | $\circ$ | $\star\star$ | $\circ$ | $\circ$ | $\star\star$ | $\star\star$ | $\star$ | $\star$ | $\star$ | |
| Transparency | | $\star\star$ | $\star\star$ | $\star$ | $\star$ | $\star\star$ | $\star\star$ | $\circ$ | $\circ$ | $\circ$ | $\circ$ |