Terascale Statistics School 2016

Multi-Variate Analysis Methods

- Web-Site: http://tmva.sourceforge.net/
- See also: "*TMVA - Toolkit for Multivariate Data Analysis , A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, H. Voss et al., arXiv:physics/0703039v5 [physics.data-an]*

Eckhard von Toerne

References

Literature:

T.Hastie, R.Tibshirani, J.Friedman, "*The Elements of Statistical Learning*", Springer 2001
C.M.Bishop, "*Pattern Recognition and Machine Learning*", Springer 2006

Software packages for Multivariate Data Analysis/Classification:

Individual classifier software:

- NeuroBayes®
- *XGBoost, https://github.com/dmlc/xgboost*
- many other packages!

"Complete" packages

StatPatternRecognition: I.Narsky, *arXiv: physics/0507143*
http://www.hep.caltech.edu/~narsky/spr.html

TMVA: Hoecker, Speckmayer, Stelzer, Therhaag, von Toerne, Voss, *arXiv: physics/0703039, included in ROOT*

Huge data analysis library available in "R": http://www.r-project.org/

universitätbonn

Contents

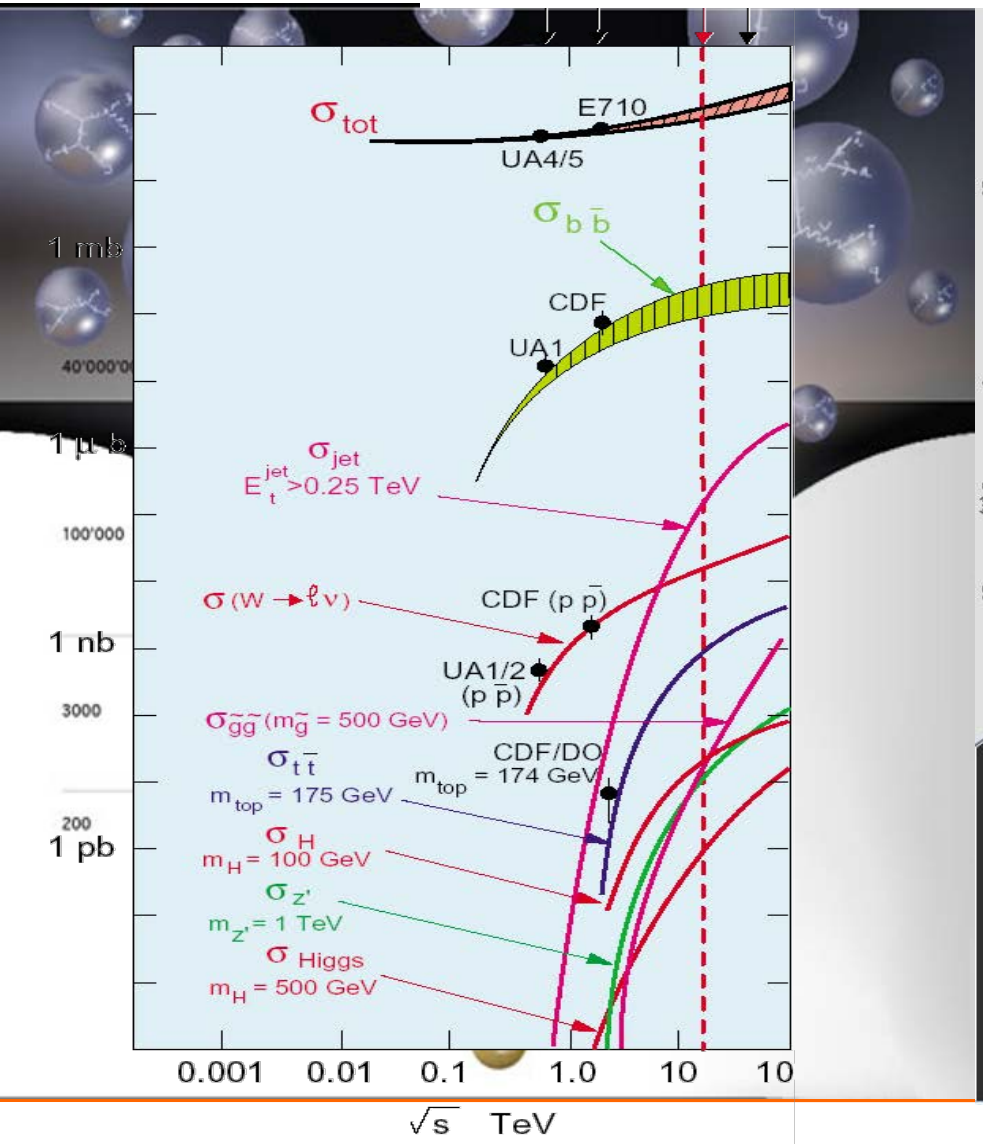
- Introduction to Multivariate Analysis
- A closer look at input data
- Overview over TMVA
- Multivariate Methods
- Using TMVA
- Tutorial

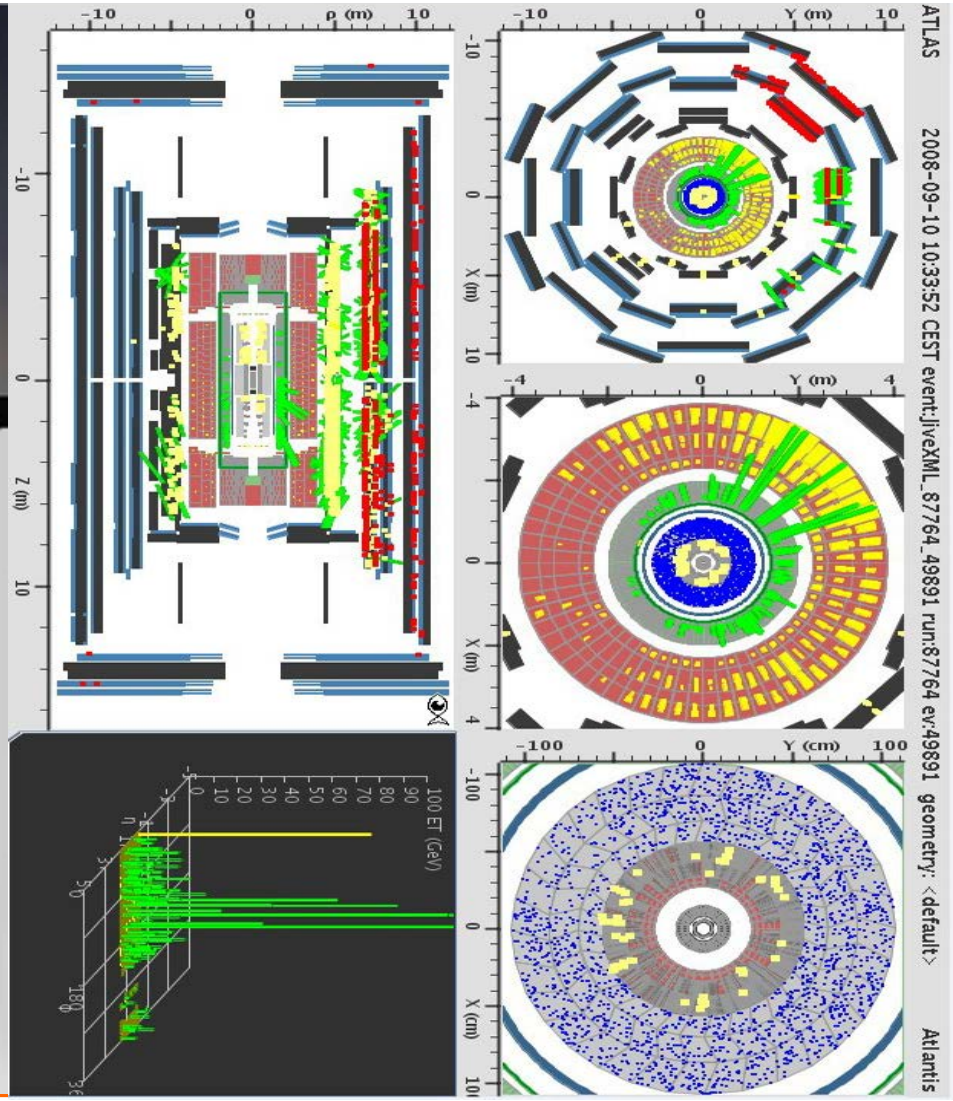
  Website: https://www.hep1.physik.uni-bonn.de/people/homepages/tmva/tmvatutorial

universitätbonn

Introduction to Multi-variate Analysis

Searches for New Physics





Searching for needles in haystacks

Can a machine learn that for you? Yes, with multi-variate techniques!

universitätbonn

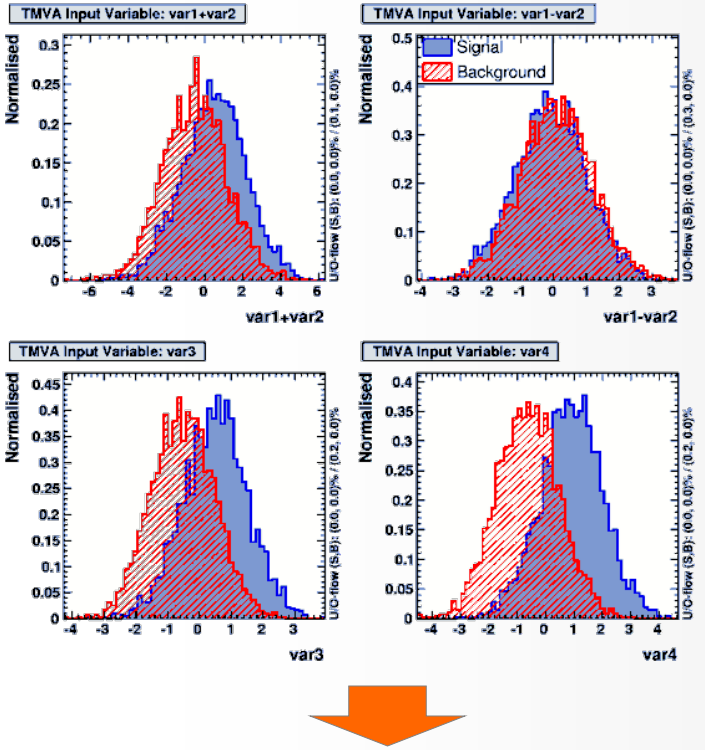
Event Classification in High-Energy Physics

- Most HEP analyses require discrimination of signal from background:
  - Event level (Higgs searches, …)
  - Cone level (Tau-vs-jet reconstruction, …)
  - Track level (particle identification, …)
  - Lifetime and flavour tagging (*b*-tagging, …)
  - Parameter estimation (*CP* violation in *B* system, …)
  - etc.
- The multivariate input information used for this has various sources
  - Kinematic variables (masses, momenta, decay angles, …)
  - Event properties (jet/lepton multiplicity, sum of charges, …)
  - Event shape (sphericity, Fox-Wolfram moments, …)
  - Detector response (silicon hits, *dE*/*dx*, Cherenkov angle, shower profiles, muon hits, …)
  - etc.
- Traditionally few powerful input variables were combined; new methods allow to use up to 100 and more variables w/o loss of classification power

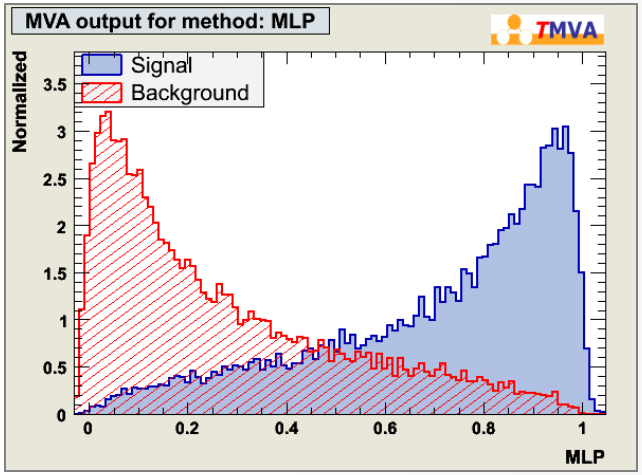
  e.g. MiniBooNE: NIMA 543 (2005), or D0 single top: Phys.Rev. D78, 012005 (2008)

What is a multi-variate analysis

- "Combine" all input variables into one output variable
- Supervised learning means learning by example: the program extracts patterns from training data
- Methods for un-supervised learning → not common in HEP and not covered in this lecture



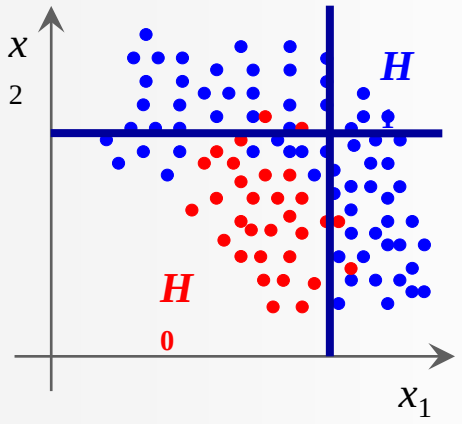
Input Variables

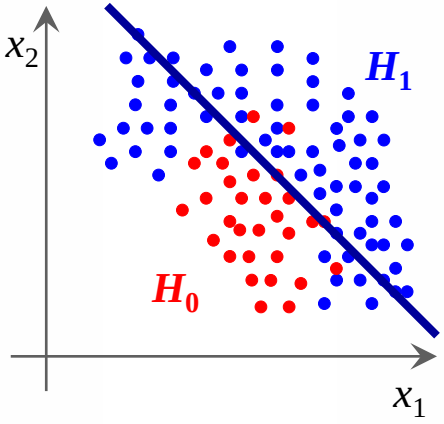


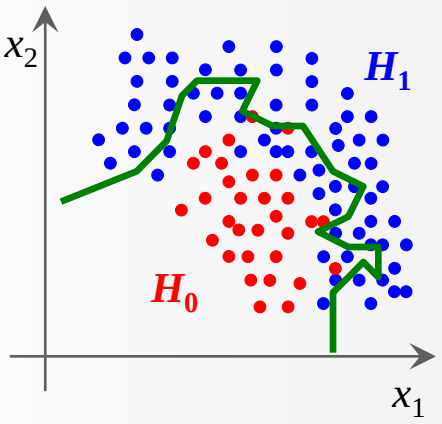
Classifier Output

Classification and Regression Tasks in HEP

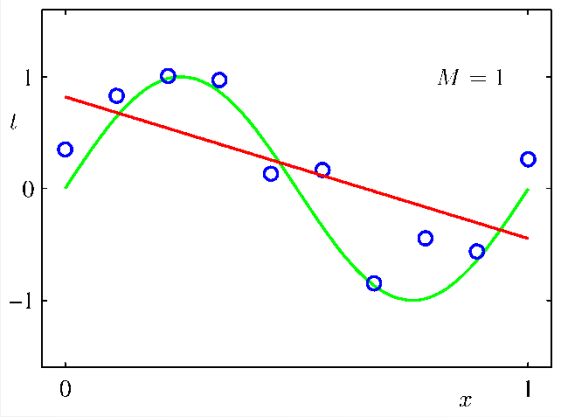
Classification of signal/background – How to find the best decision boundary?

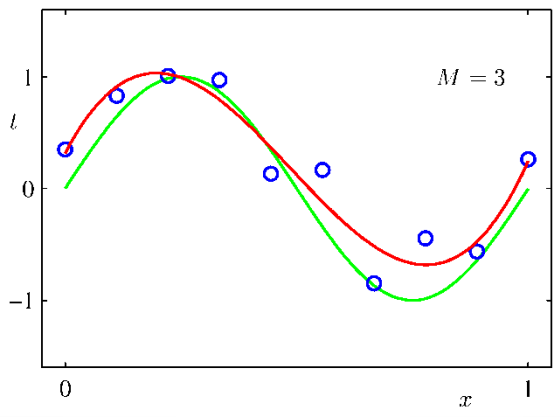


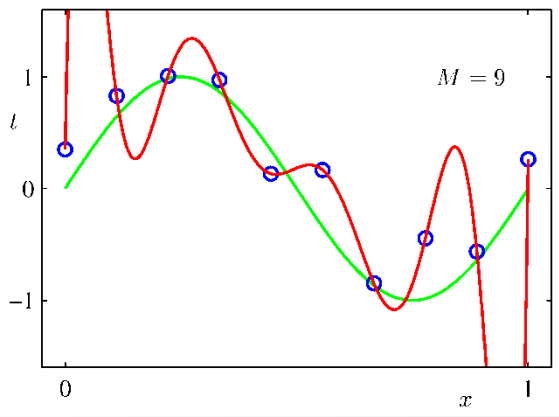




Regression – How to determine the correct model?



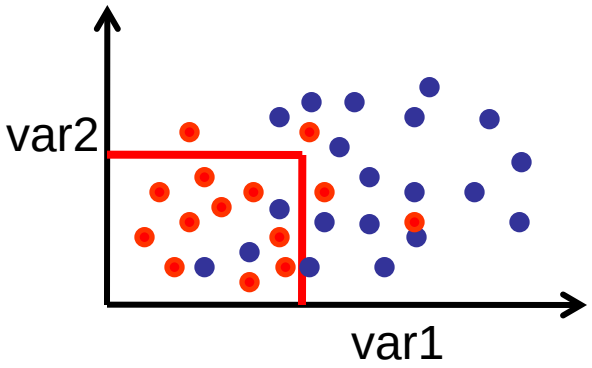




Can a machine learn that for you? Yes, with TMVA!

Multi-variate methods: Cuts

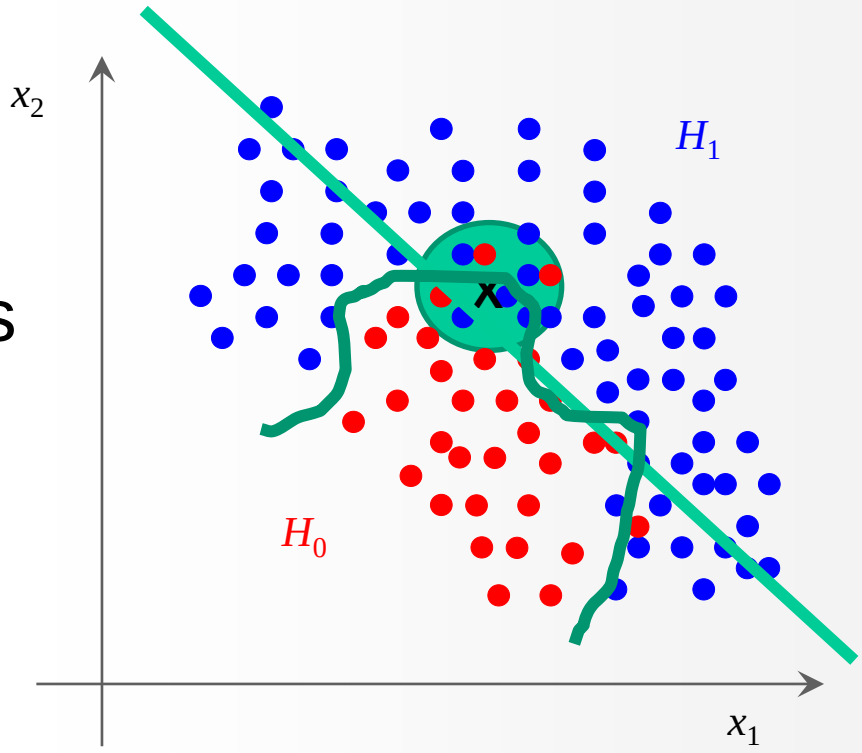
- Simplest multi-variate classification method: **Cuts**
- Signal region is defined by a series of cuts:
  - Var1 > x1
  - Var2 < x2 ...



- However not necessarily the easiest approach
  - Cuts have difficulties if variables have low separation power and many variables are involved
- Whenever cut selections are employed, more sophisticated multi-variate methods may also work

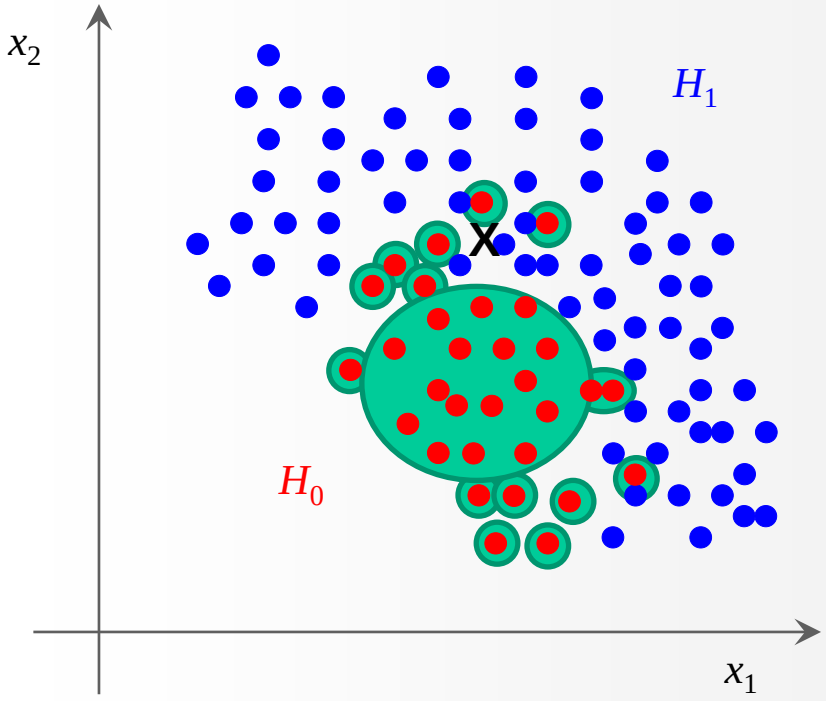
MVA methods

- Kernel PDE (probablility density estimators)
- functional approaches (Linear, Likelihoods, …)
- General methods:
  - Neural nets,
  - Boosted Decision trees
  - Support Vector machines



Overtraining

- If the MVA follows statistical fluctuations of input training data → performance will not be reproducable on independent training data.



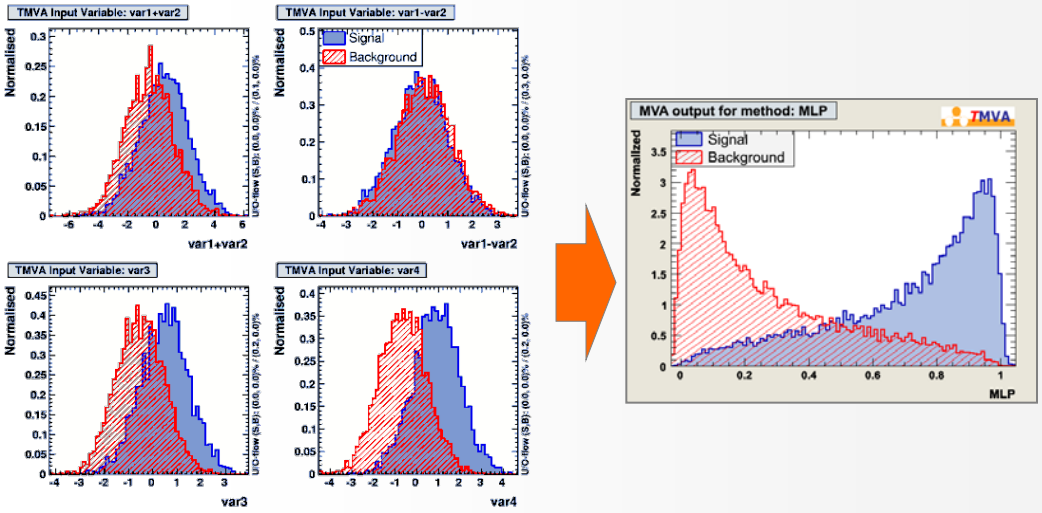
Typical multi-variate analysis steps

- Choice of input variables
- Define preselection

} physics input is crucial

- Choice of MVA method
- Training the MVA method using samples with known signal/background
- Choice of working point

}



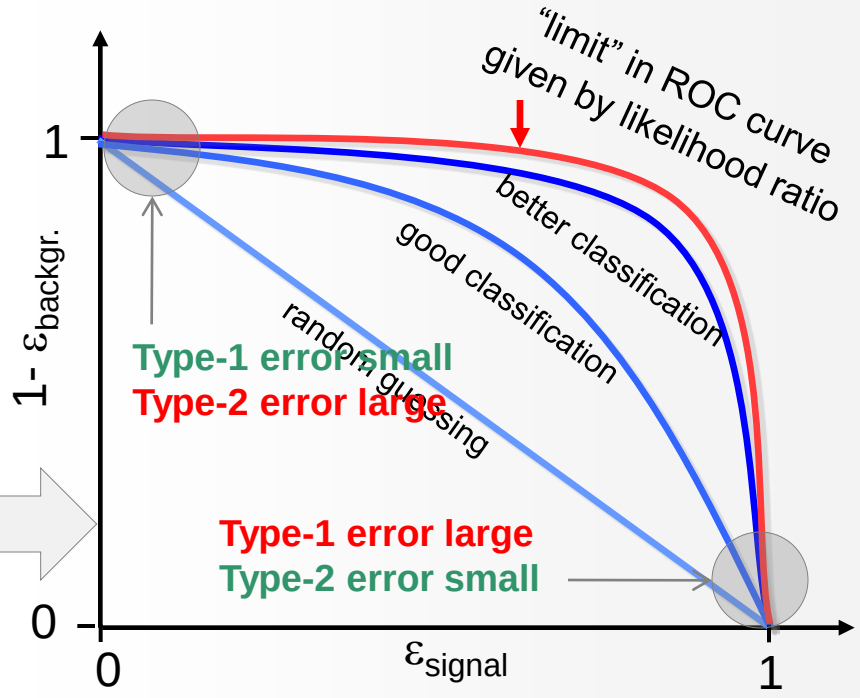
Neyman-Pearson Lemma

$$\text{Likelihood Ratio}: \quad y(x) = \frac{P(x|S)}{P(x|B)}$$

Neyman-Peason:

The Likelihood ratio used as "selection criterion" y(x) gives for each selection efficiency the best possible background rejection.

i.e. it maximises the area under the "*Receiver Operation Characteristics*" (ROC) curve



Varying y(x)>"cut" moves the working point (efficiency and purity) along the ROC curve

How to choose "cut"? → need to know prior probabilities (S, B abundances)

- Measurement of signal cross section: maximum of S/√(S+B) or equiv. √(ε·p)
- Discovery of a signal : maximum of S/√(B)
- Precision measurement: high purity (p)
- Trigger selection: high efficiency (ε)

universitätbonn

A Closer Look at Input Data

General data properties

- Variables may be statistically (un-)correlated
- Signal and/or Background may cover full volume, partial volume, or are only found on hypersurfaces.
- Variables may have spikes, steps, tails, poles
- One or many connected regions
- Number of variables

Highly-Dimensional Data Spaces

- Data Vector $(x_1, x_2,\ldots,x_N)$
- Math is same if N=3 or N=30 … subtle changes occur as N incr…
  - Most of the Volume … sphere is in its outermost shell … or
  - Volume of … N-sphere
  - V(N … (R=side length)
  - … e) = $c_N * R^N$ (R=radius)
  - … 14, c_3=4.19, c_4=4.93, c_5=5.26, c_6=5.10, c_7=4.65, c_8=3.96

Curse of Dimensionality

$\lim_{N\to\infty} c_N = 0$

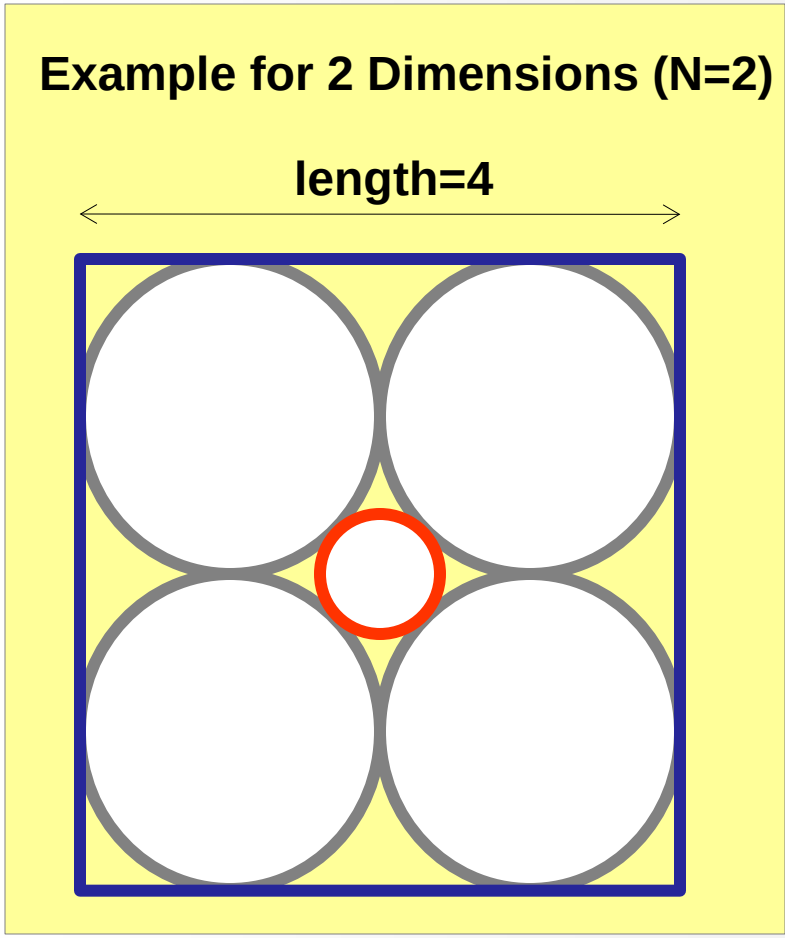
(most of the cube's volume is in its corners)

Exercise:
Fill an N-cube of length 4 with R=1 N-spheres in a cubic lattice

- How many spheres with R=1 fit into cube (using cubic lattice)?

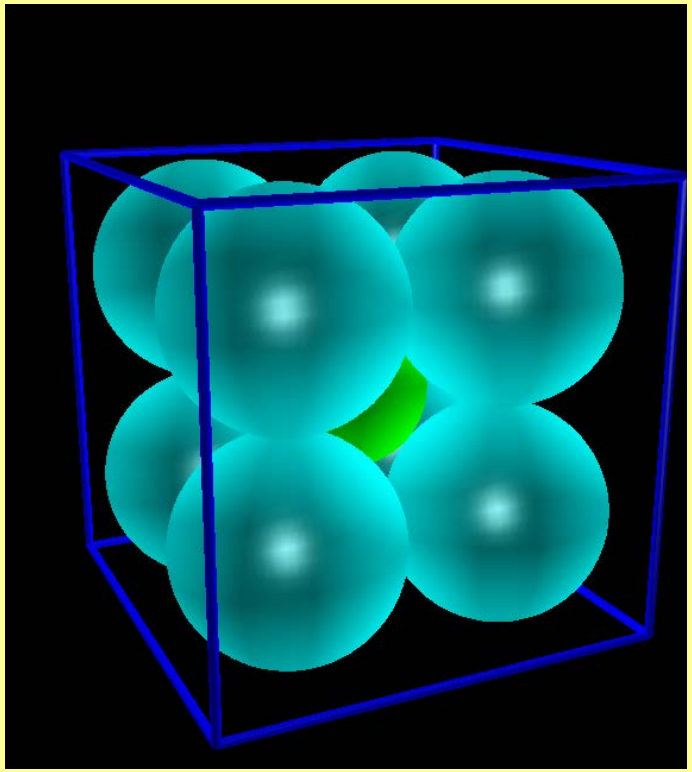
→ 2^N

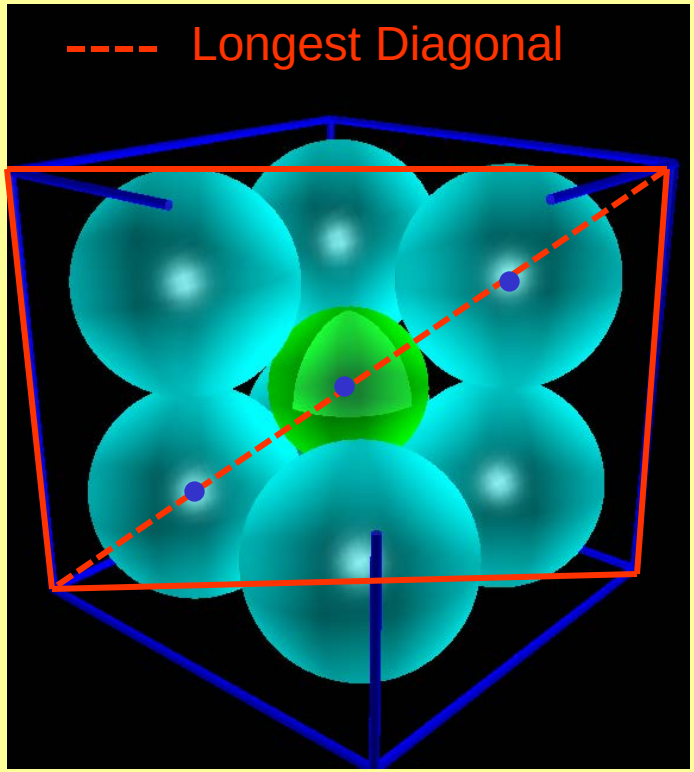
- Now insert into the center of the cube another sphere such that it touches the surrounding spheres. What is its radius?



Exercise:
Fill an N-cube of length 4 with R=1 N-spheres in a cubic lattice

Example for 3 Dimensions



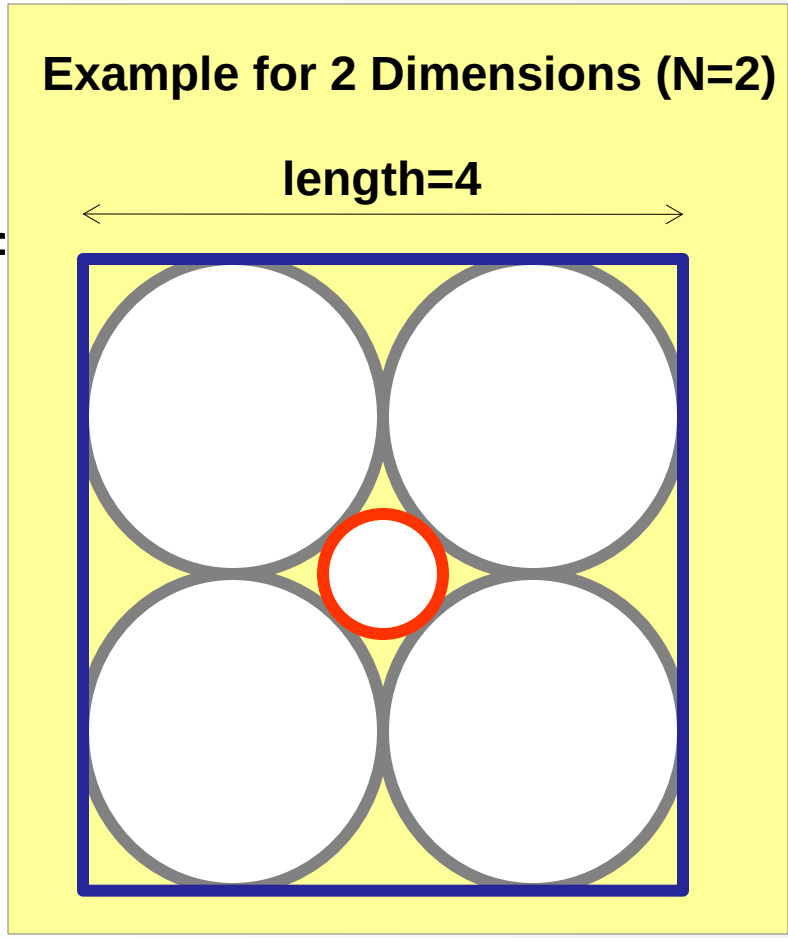


Exercise:
Fill an N-cube of length 4 with R=1 N-spheres in a cubic lattice

- How many spheres with R=1 fit in (using cubic lattice)?

→ 2^N

- Now insert into the center of the cube another sphere such that it touches the surrounding spheres. What is its radius? → **Sqrt(N)-1**
- **For large N, the inner sphere exceeds the cube!**



universitätbonn

Toolkit for MultiVariate Analysis (TMVA)

What is TMVA?

- TMVA: not just a collection of MVA methods for supervised learning, but also provides…
  - a common interface for all MVA techniques
  - a common interface for classification and regression
  - easy training and testing of all methods on the same datasets
    - consistent evaluation and comparison
    - common data preprocessing
  - a complete user analysis framework and examples
  - embedded in ROOT
  - creation of standalone C++ classes (ROOT independent)
  - an understandable Users Guide
  - "*TMVA - Toolkit for Multivariate Data Analysis , A. Hoecker, ..., E.v.Toerne et al.,* *arXiv:physics/0703039v5 [physics.data-an]*

http://tmva.sourceforge.net

universitätbonn

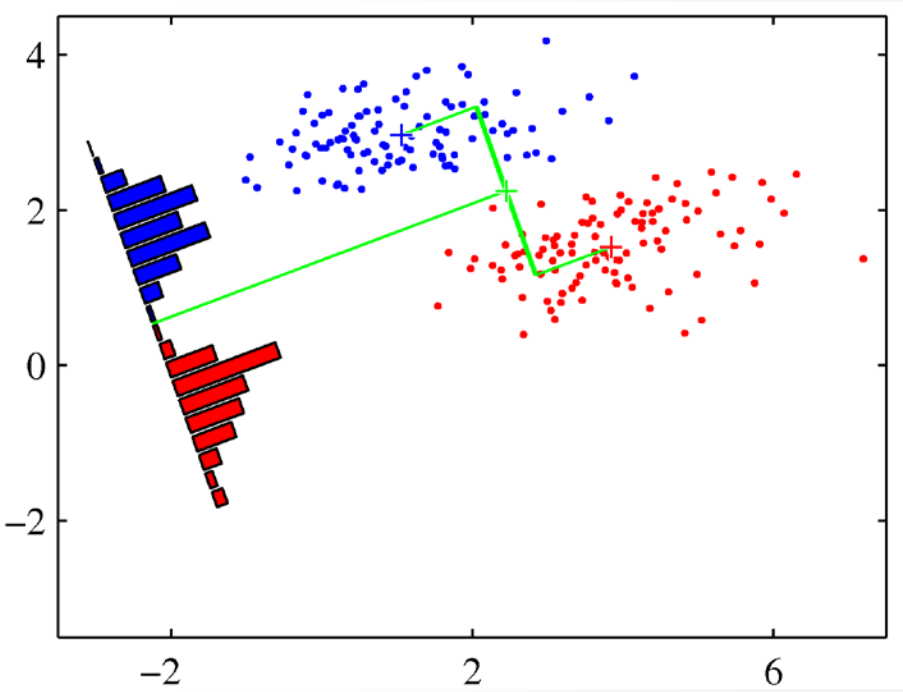
The TMVA Classifiers

Cuts, Fisher, Likelihood, Functional Discriminant, kNN, Neural Networks, Support Vector Machine, Boosted Decision Trees, Rule Fit…

Fisher (linear discriminant)

- Fisher's discriminant is a linear model which projects the data on the (hyper)plane of best separation

- Advantages:
  - easy to understand
  - robust, fast
- Disadvantages:
  - not very flexible
  - poor performace in complex settings

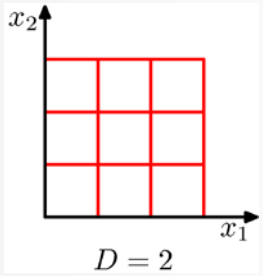


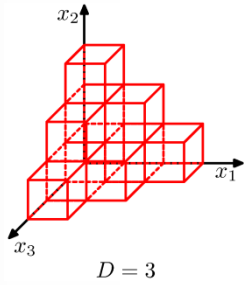
Performance		Speed		Robustness		Curse of Dim.	Transparency	Regression	
No/linear correlations	Nonlinear correlations	Training	Response	Overtraining	Weak input vars			1D	multi D
☺	☹	☺	☺	☺	☺	☺	☺	☺	☹

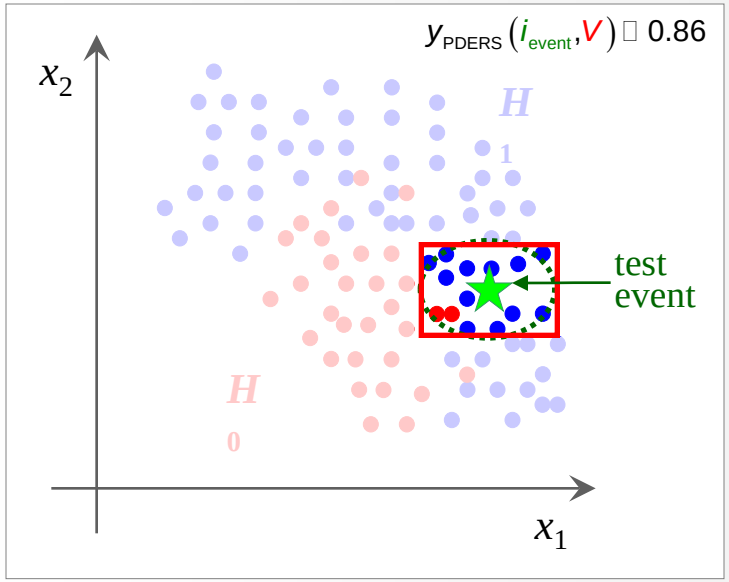
Multidimensional PDE

- Idea: Estimate the multidimensional probability densities by counting the events of each class in a predefined or adaptive volume

- Advantages:
  - all correlations taken into account
- Disadvantages:
  - „curse of dimensionality“







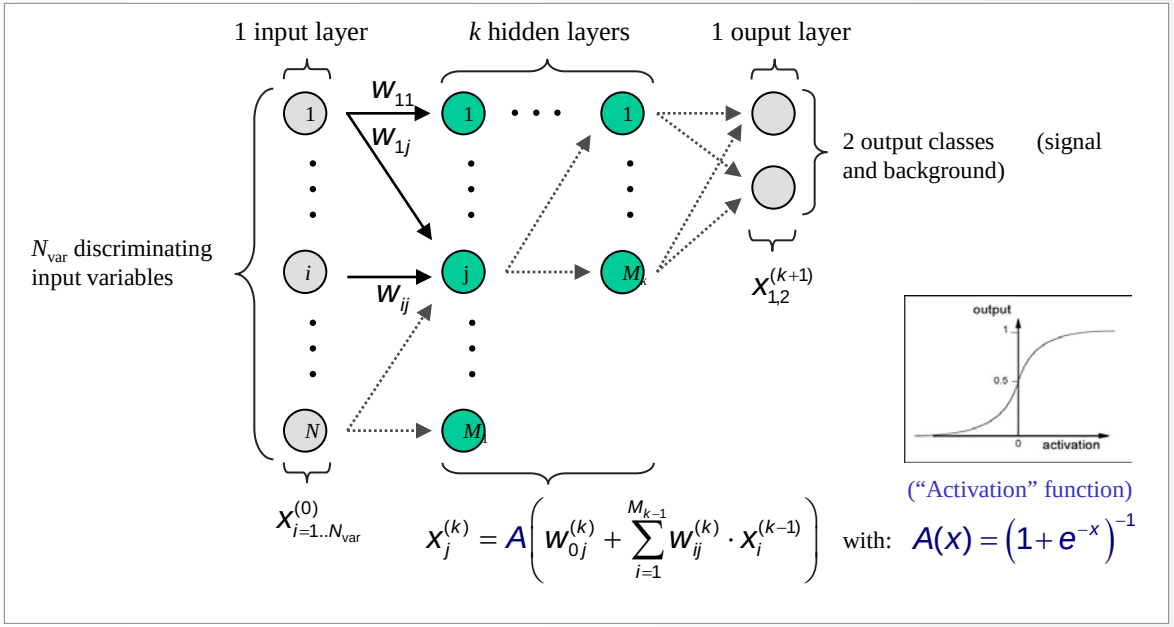
Performance		Speed		Robustness		Curse of Dim.	Transparency	Regression	
No/linear correlations	Nonlinear correlations	Training	Response	Overtraining	Weak input vars			1D	multi D
☺	☺	☺	😐/☹	😐	☹	☹	😐	☹/☺	☹/☺

Artificial Neural Networks

- Modelling of arbitrary nonlinear functions as a nonlinear combination of simple „neuron activation functions“

- Advantages:
  - very flexible, no assumption about the function necessary
- Disadvantages:
  - „black box“
  - needs tuning
  - seed dependent

Feed-forward Multilayer Perceptron



Performance		Speed		Robustness		Curse of Dim.	Transparency	Regression	
No/linear correlations	Nonlinear correlations	Training	Response	Overtraining	Weak input vars			1D	multi D
☺	☺	😐	☺	😐	😐	😐	☹	☺	☺

How to choose a method?

- If you have a training sample with only few events?
  - → Number of „parameters“ must be limited
  - → Use Linear classifier or FDA, small BDT, small MLP
- Variables are uncorrelated (or only linear corrs) → likelihood
- I just want something simple → use Cuts, LD, Fisher
- Methods for complex problems → use BDT, MLP, SVM

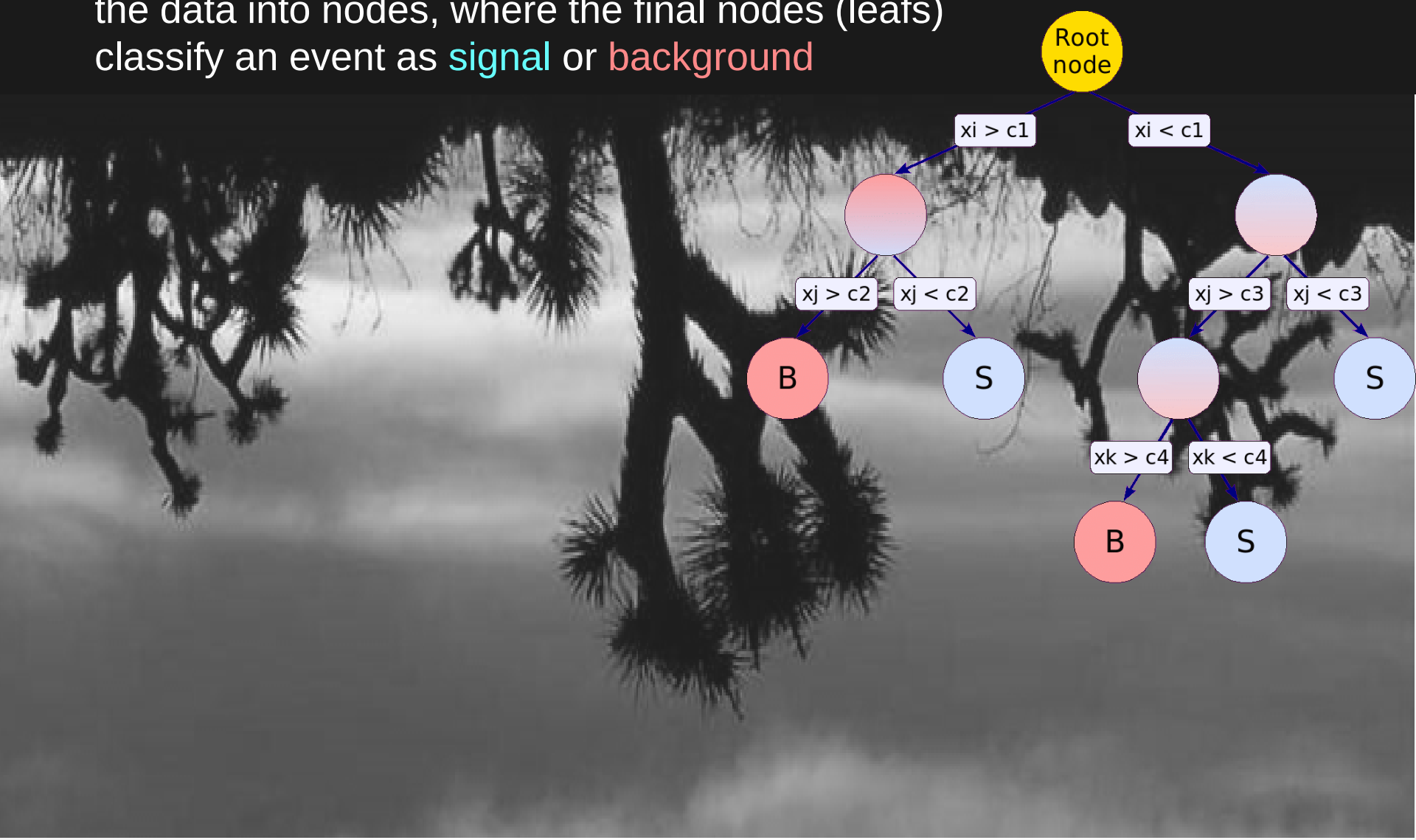
List of acronyms:
BDT = boosted decision tree, see manual page 103
ANN = articifical neural network
MLP = multi-layer perceptron, a specific form of ANN, also the name of our flagship ANN, manual p. 92
FDA = functional discriminant analysis, see manual p. 87
LD = linear discriminant , manual p. 85
SVM = support vector machine, manual p. 98 , SVM currently available only for classification
Cuts = like in “cut selection“, manual p. 56
Fisher = Ronald A. Fisher, classifier similar to LD, manual p. 83

universitätbonn

Boosted Decision Trees

Decision Trees

Decision Tree: Sequential application of cuts splits the data into nodes, where the final nodes (leafs) classify an event as signal or background

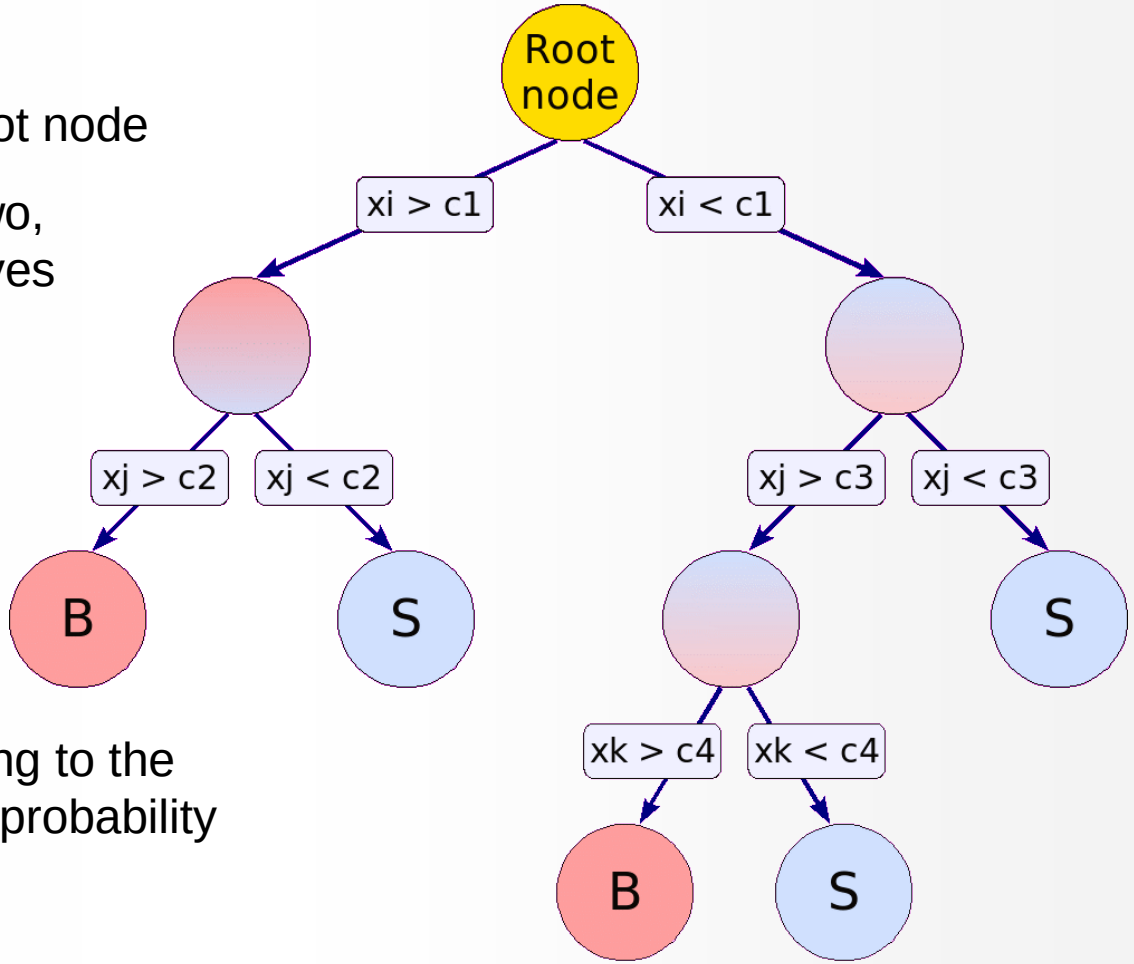


Growing a Decision Tree

start with training sample at the root node

split training sample at node into two, using a cut in the variable that gives best separation gain

continue splitting until:
- minimal #events per node
- maximum number of nodes
- maximum depth specified

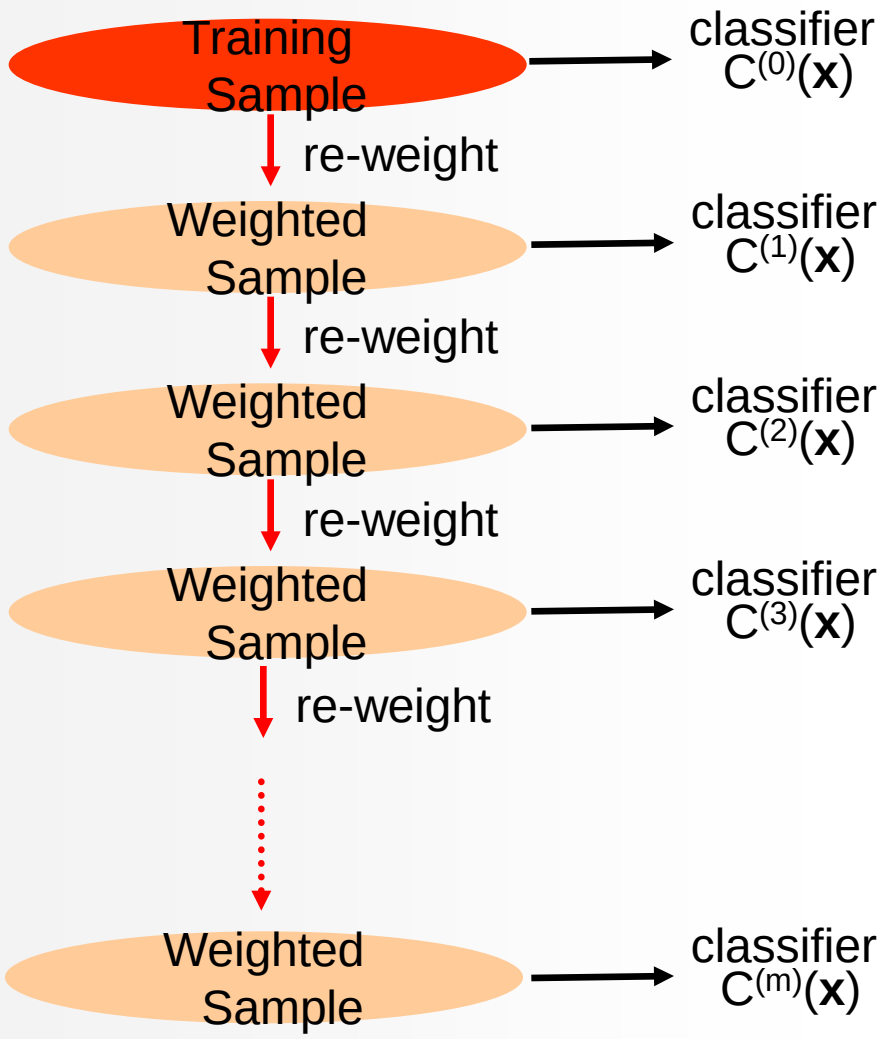


leaf-nodes classify as S,B according to the majority of events or give a S/B probability

Why no multiple branches (splits) per node ?

→ Fragments data too quickly; also: multiple splits per node = series of binary node splits

Adaptive Boosting (AdaBoost)



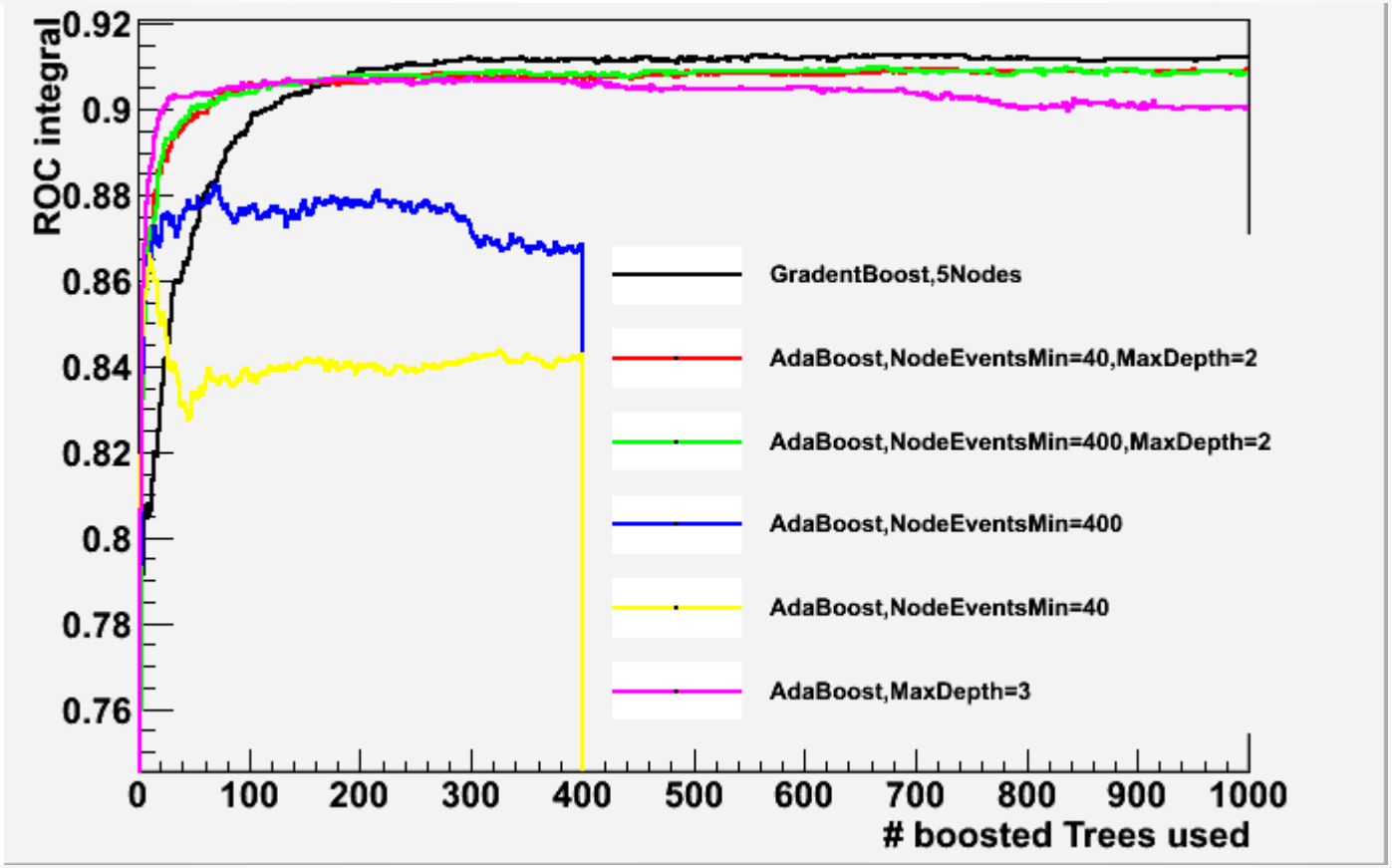
AdaBoost re-weights events misclassified by previous classifier by:

$$\frac{1-f_{err}}{f_{err}} \quad \text{with:}$$

$$f_{err} = \frac{\text{misclassified events}}{\text{all events}}$$

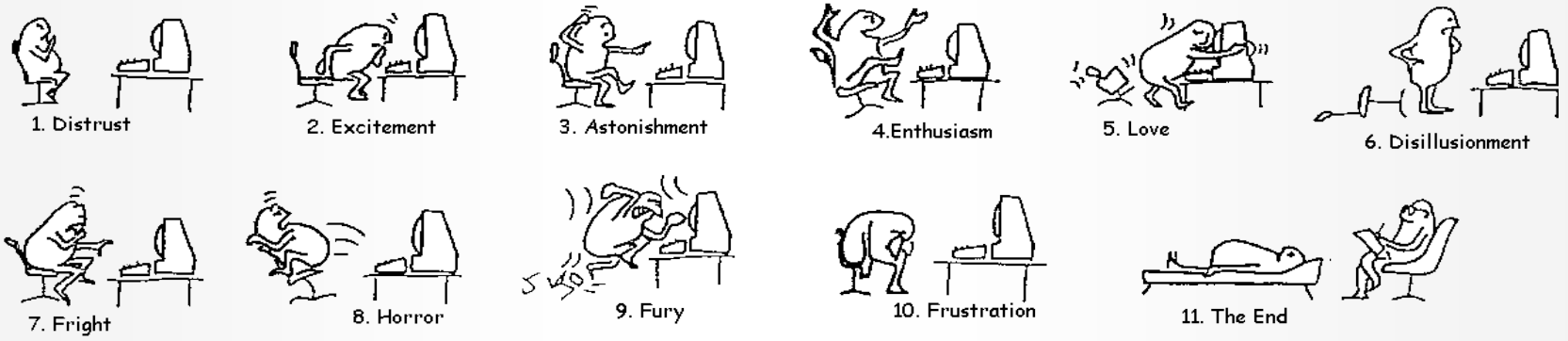
AdaBoost weights the classifiers also using the error rate of the individual classifier according to:

$$y(x) = \sum_{i}^{N_{Classifier}} \log\left(\frac{1-f_{err}^{(i)}}{f_{err}^{(i)}}\right) C^{(i)}(x)$$



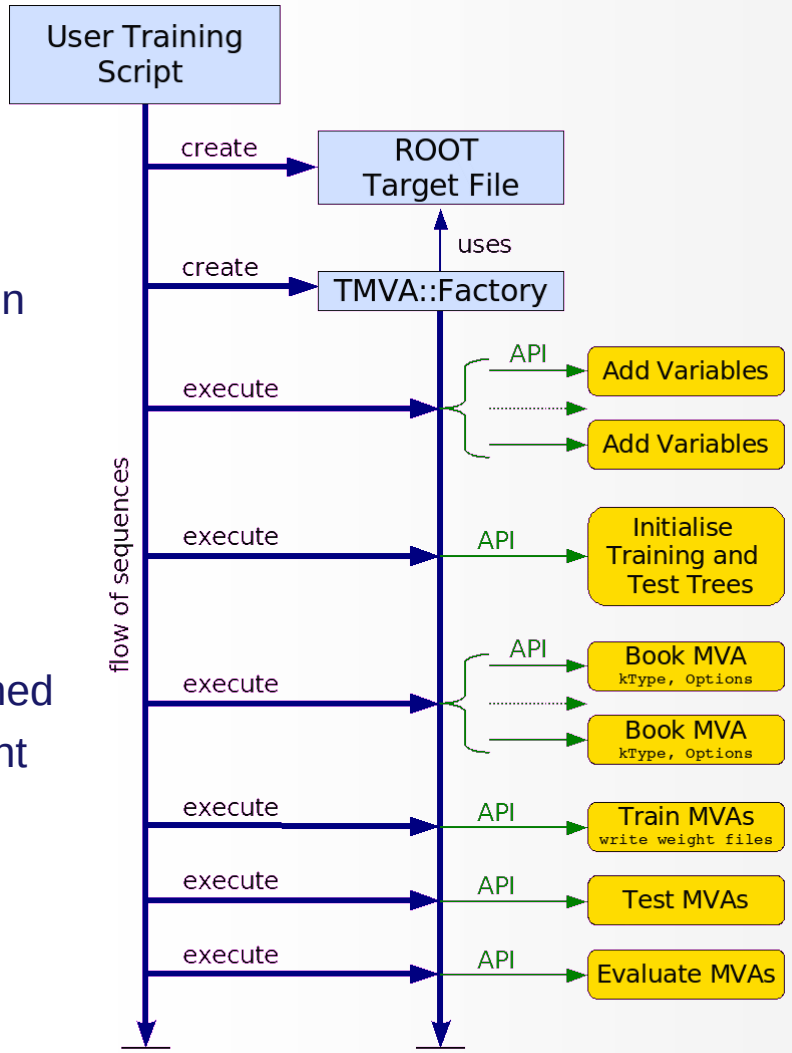
Boosting seems to work best on “weak” classifiers (i.e. small, dumb trees)
Tuning (tree building) parameter settings are important
For good out of the box performance: Large numbers of very small trees

Using TMVA


1. Distrust
2. Excitement
3. Astonishment
4.Enthusiasm
5. Love
6. Disillusionment
7. Fright
8. Horror
9. Fury
10. Frustration
11. The End

The TMVA Workflow

- Training:
  - Classification:
    Learn the features of the different event classes from a sample with known signal/background composition
  - Regression:
    Learn the functional dependence between input variables and targets
- Testing:
  - Evaluate the performance of the trained classifier/regressor on an independent test sample
  - Compare different methods
- Application:
  - Apply the classifier/regressor to real data



Flexibility and customization

- Data input formats: ROOT TTree or ASCII
- Supports selection of any subset or combination or function of available variables (including all TMath functions)
- Supports independent signal/bkg pre-selection cuts
- Supports global event weights for signal/bkg input files
- Supports use of any input variable as individual event weight
- Supports various methods for splitting into training and test samples
  - block, random, periodic, user defined trees

….

A complete TMVA training/testing session

```
void TMVAnalysis( )
{
  TFile* outputFile = TFile::Open( "TMVA.root", "RECREATE" );

  TMVA::Factory *factory = new TMVA::Factory( "MVAnalysis", outputFile,"!V");

  TFile *input = TFile::Open("tmva_example.root");

  factory->AddVariable("var1+var2", 'F');
  factory->AddVariable("var1-var2", 'F');   //factory->AddTarget("tarval", 'F');

  factory->AddSignalTree ( (TTree*)input->Get("TreeS"), 1.0 );
  factory->AddBackgroundTree ( (TTree*)input->Get("TreeB"), 1.0 );
  //factory->AddRegressionTree ( (TTree*)input->Get("regTree"), 1.0 );
  factory->PrepareTrainingAndTestTree( "", "",
  "nTrain_Signal=200:nTrain_Background=200:nTest_Signal=200:nTest_Background=200:!V" );

  factory->BookMethod( TMVA::Types::kLikelihood, "Likelihood",
  "!V:!TransformOutput:Spline=2:NSmooth=5:NAvEvtPerBin=50" );
  factory->BookMethod( TMVA::Types::kMLP, "MLP",
  "!V:NCycles=200:HiddenLayers=N+1,N:TestRate=5" );

  factory->TrainAllMethods();  // factory->TrainAllMethodsForRegression();
  factory->TestAllMethods();
  factory->EvaluateAllMethods();
  outputFile->Close();
  delete factory;
}
```

Create Factory

Add variables/ targets

Initialize Trees

Book MVA methods

Train, test and evaluate

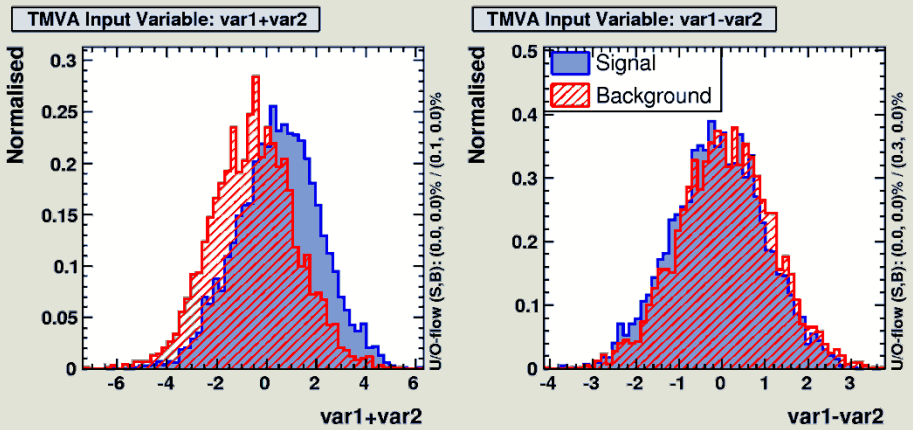
The method option string

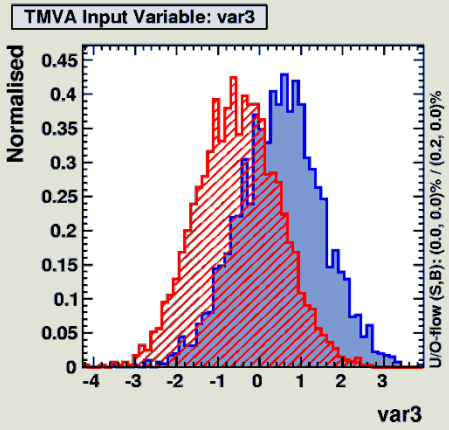
```
void TMVAnalysis( )
{
  TFile* outputFile = TFile::Open( "TMVA.root", "RECREATE" );

  TMVA::Factory *factory = new TMVA::Factory( "MVAnalysis", outputFile,"!V");

  TFile *input = TFile::Open("tmva_example.root");

  factory->AddVariable("var1+var2", 'F');
  factory->AddVariable("var1-var2", 'F');   //factory->A

  factory->AddSignalTree ( (TTree*)input->Get("Tree
  factory->AddBackgroundTree ( (TTree*)input->Get(
  //factory->AddRegressionTree ( (TTree*)input->Get
  factory->PrepareTrainingAndTestTree( "", "",
  "nTrain_Signal=200:nTrain_Background=200:nTest_Signal=200:nTest_Background=200:!V" );

  factory->BookMethod( TMVA::Types::kLikelihood, "Likelihood",
  "!V:!TransformOutput:Spline=2:NSmooth=5:NAvEvtPerBin=50" );
  factory->BookMethod( TMVA::Types::kMLP, "MLP",
  "!V:NCycles=200:HiddenLayers=N+1,N:TestRate=5" );

  factory->TrainAllMethods();  // factory->TrainAllMethodsForRegression();
  factory->TestAllMethods();
  factory->EvaluateAllMethods();
  outputFile->Close();
  delete factory;
}
```

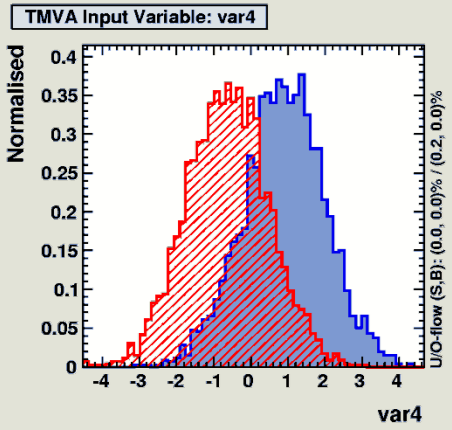
pay attention to the option string:
"!V:NCycles=200:HiddenLayers=N+1,N…"
List of possible options in TMVA manual
arXiv:0703039v5

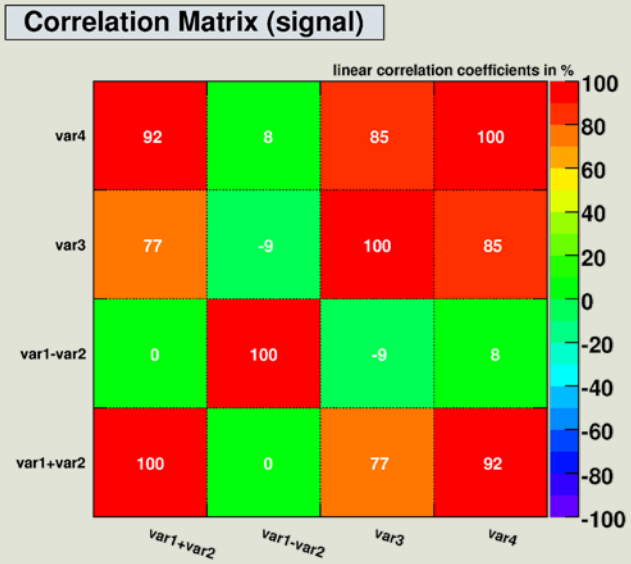
A Toy Example (idealized)

- Use data set with 4 linearly correlated Gaussian distributed variables:









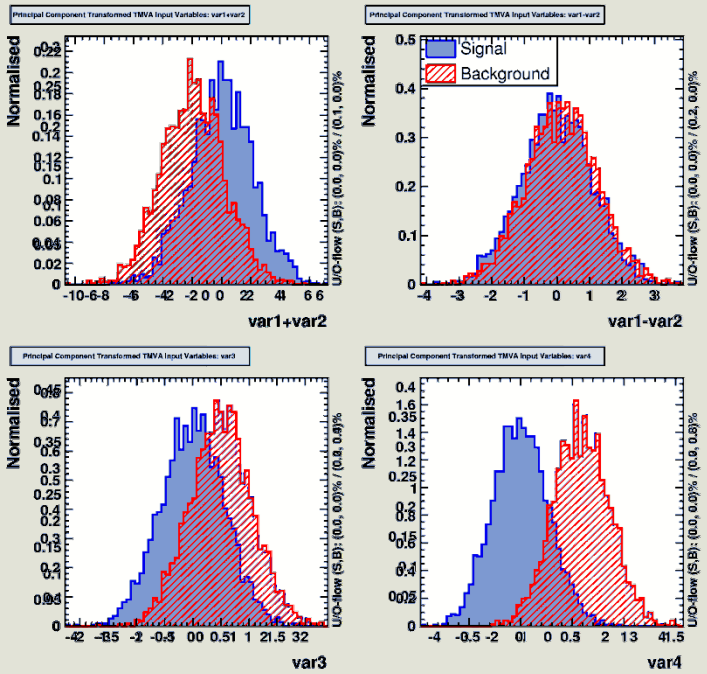
Rank : Variable : Separation

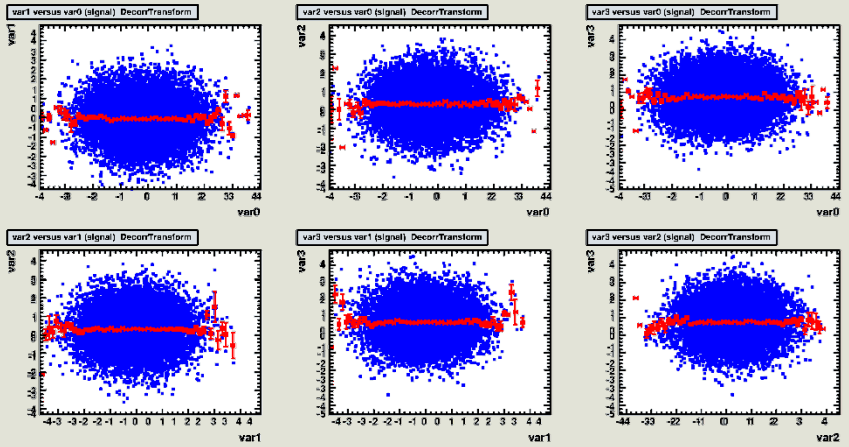
---

1 : var4 : 0.606
2 : var1+var2 : 0.182
3 : var3 : 0.173
4 : var1-var2 : 0.014

Preprocessing the Input Variables

- Decorrelation of variables before training is useful for *this* example

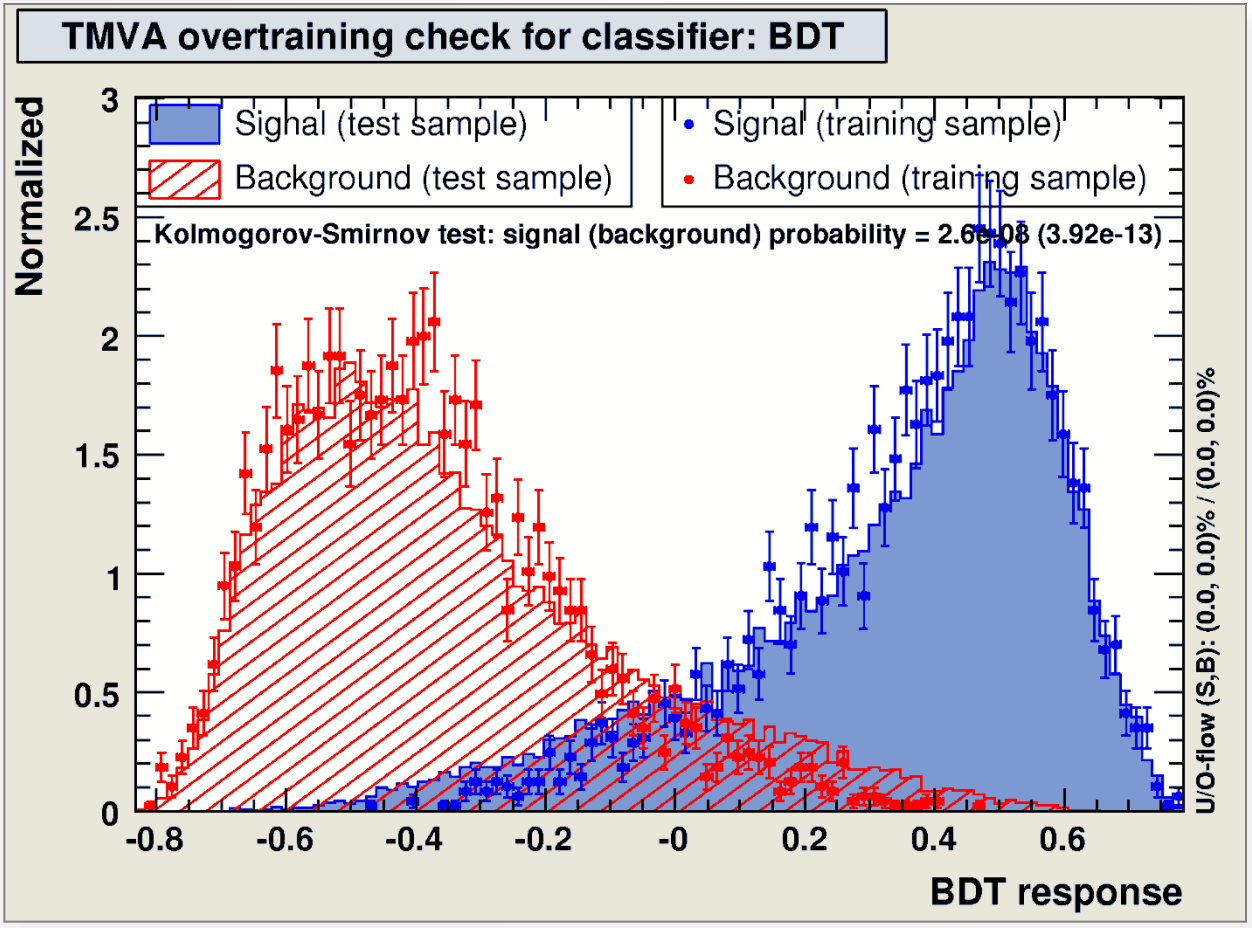




- Note that in cases with non-Gaussian distributions and/or nonlinear correlations decorrelation may do more harm than any good

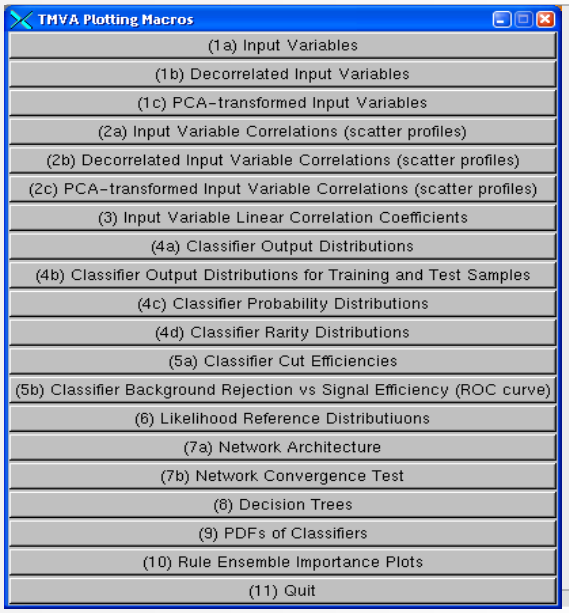
Evaluating the Classifier Training (II)

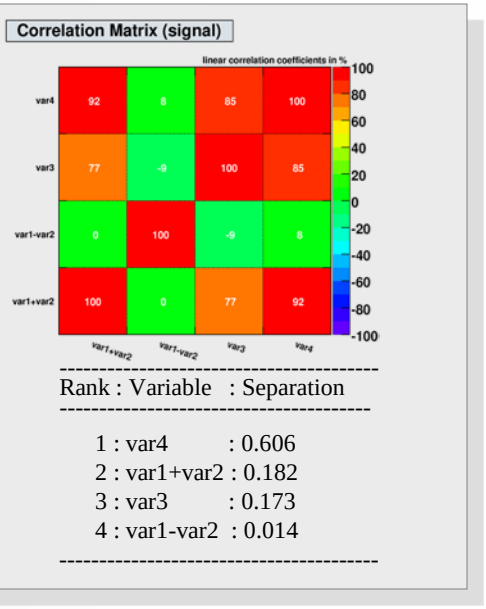
- Check for overtraining: classifier output for test *and* training samples …



- Remark on **overtraining**
  - Occurs when classifier training has too few degrees of freedom because the classifier has too many adjustable parameters for too few training events
  - Sensitivity to overtraining depends on classifier: *e.g.*, Fisher weak, BDT strong
  - Compare performance between training and test sample to detect overtraining
  - Actively counteract overtraining: *e.g.*, smooth likelihood PDFs, prune decision trees, …

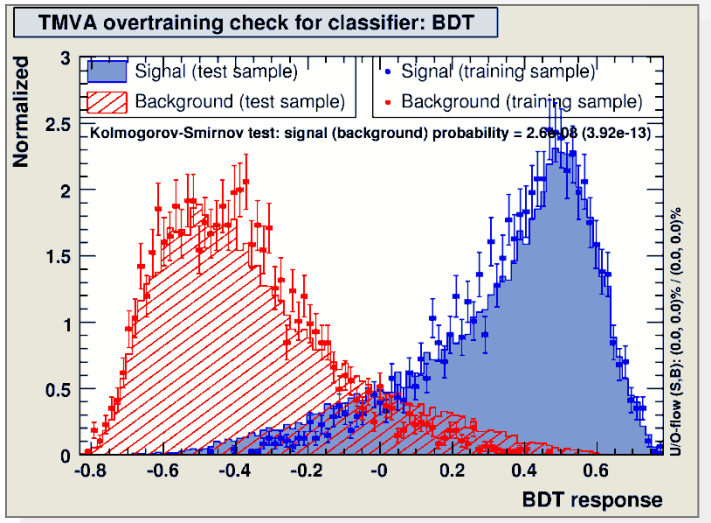
The TMVA GUI

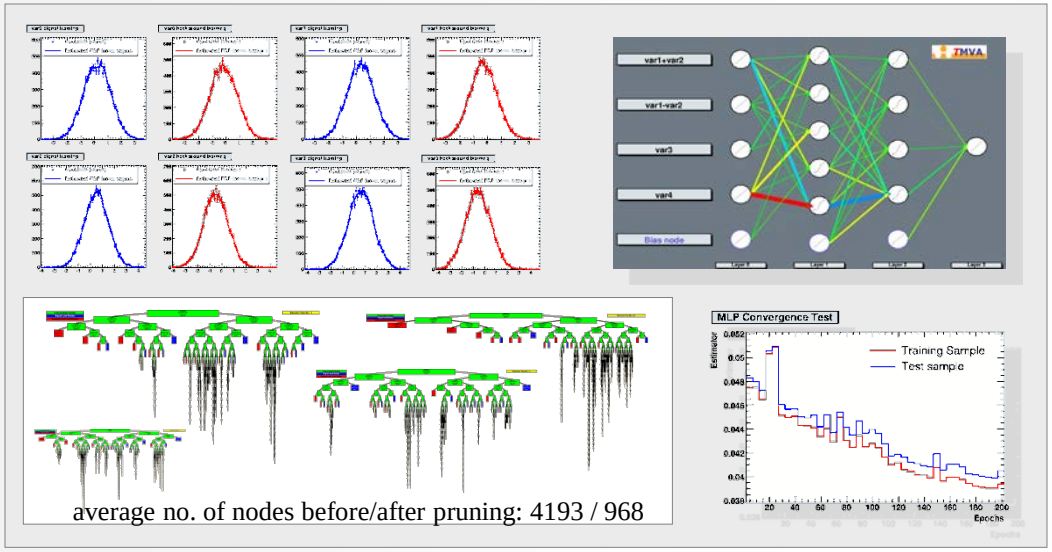




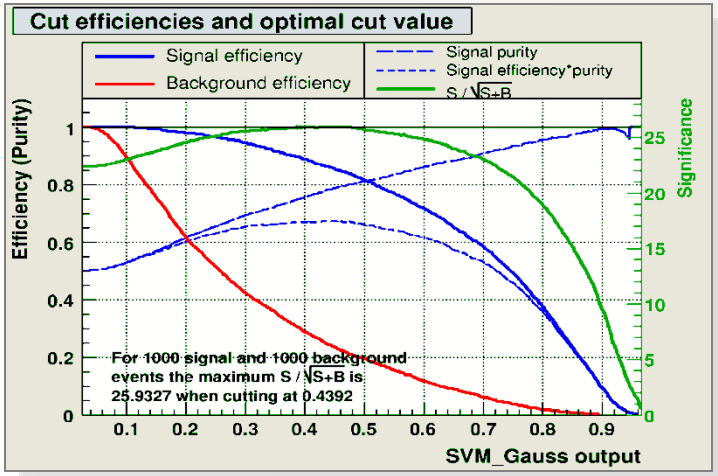
Rank : Variable : Separation

1 : var4 : 0.606
2 : var1+var2 : 0.182
3 : var3 : 0.173
4 : var1-var2 : 0.014



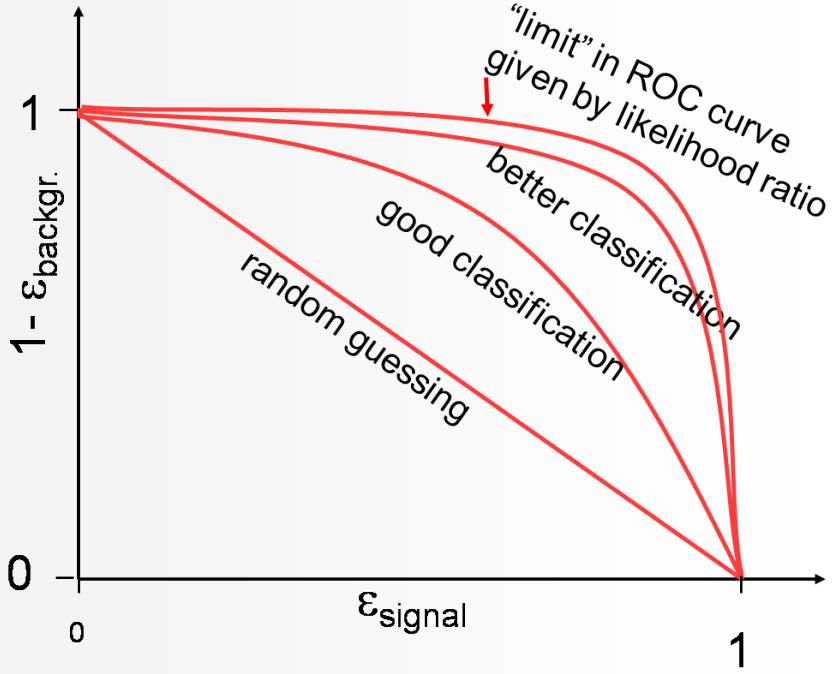


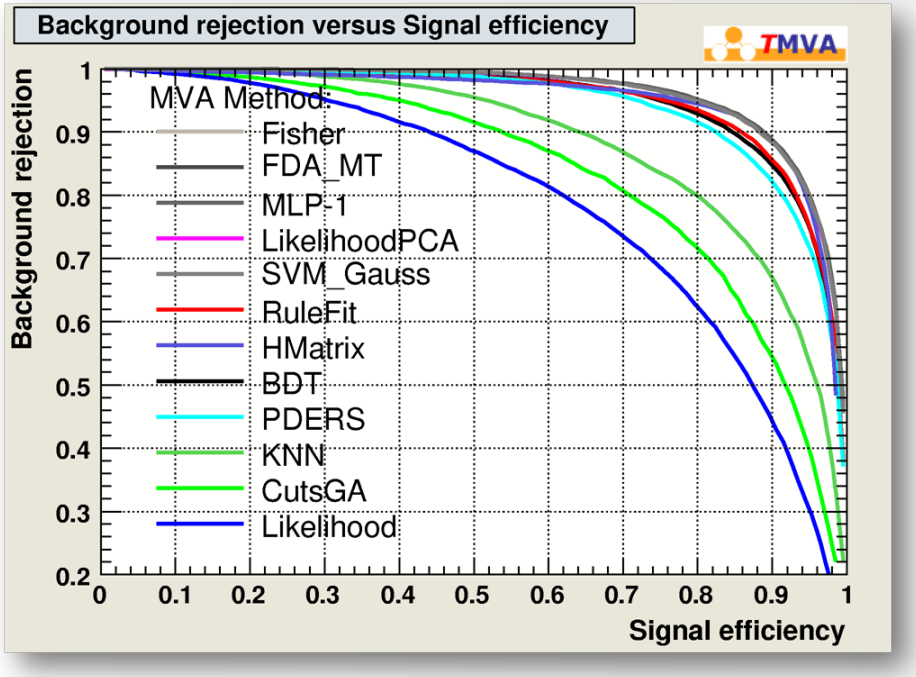
average no. of nodes before/after pruning: 4193 / 968



The ROC curve

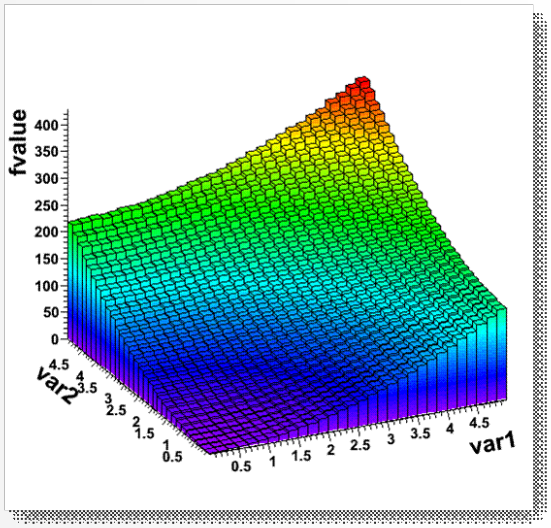
- Easy comparison of classifier performance through the ROC* curve is provided after the training/testing

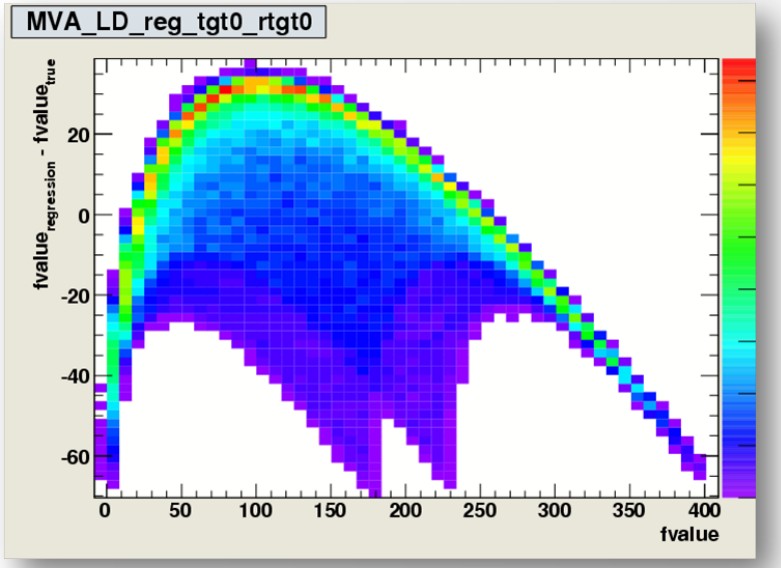


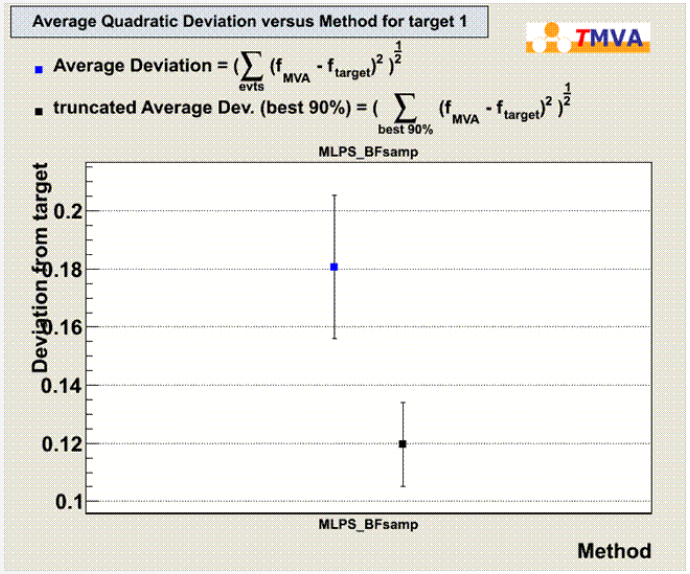


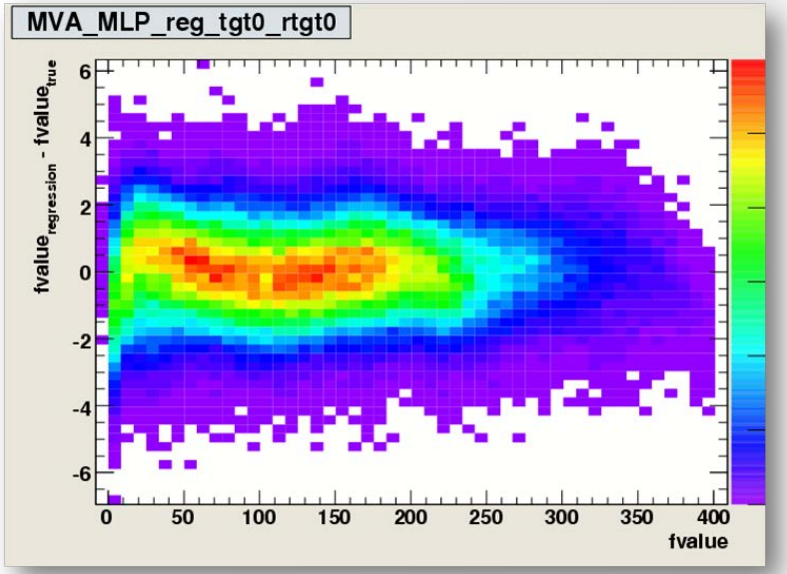
* **ROC**=**R**eceiver **O**perations **C**haracteristics

TMVA GUI (Regression)









Non-HEP applications and Data Science

test

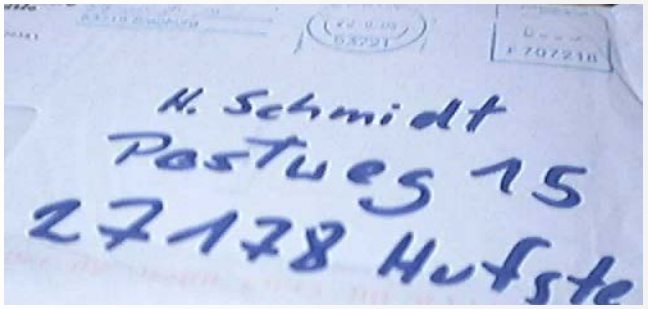

kaggle
KAGGLE
Start compe
The Home of Data Science
ENERGY INDUSTRY SPECIALISTS • PREDICTIVE MODELING SOLUTIONS •
COMPETITIONS FOR DATA SCIENTISTS • TUTORIALS •
UNIVERSITY COMPETITIONS • DATA SCIENCE JOBS BOARD
HIRING DATA SCIENTISTS?
POST TO OUR JOBS BOARD ›
FOR CUSTOMERS

universitätbonn

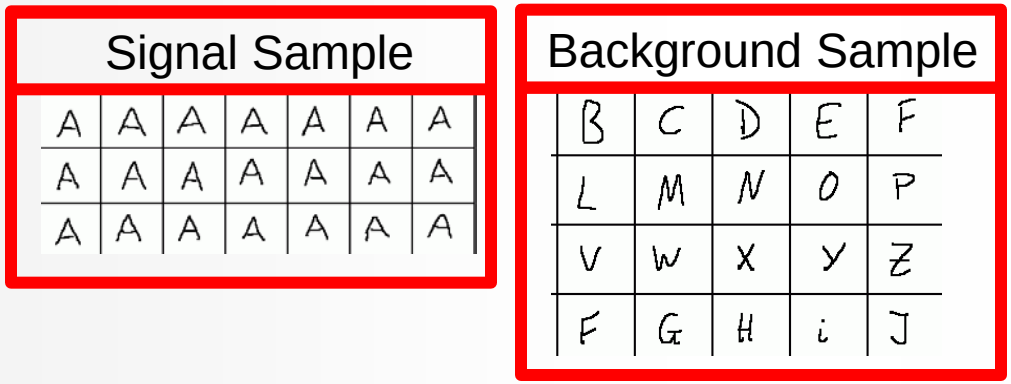
A simple non-HEP example for a multi-variate classification task

Pattern Recognition of handwritten digits

- Automatic reading of handwritten digits for Zip-code processing in mail services (Postleitzahlerkennung)



- One MVA methods for each digit: one digit is Signal, everything else is Background



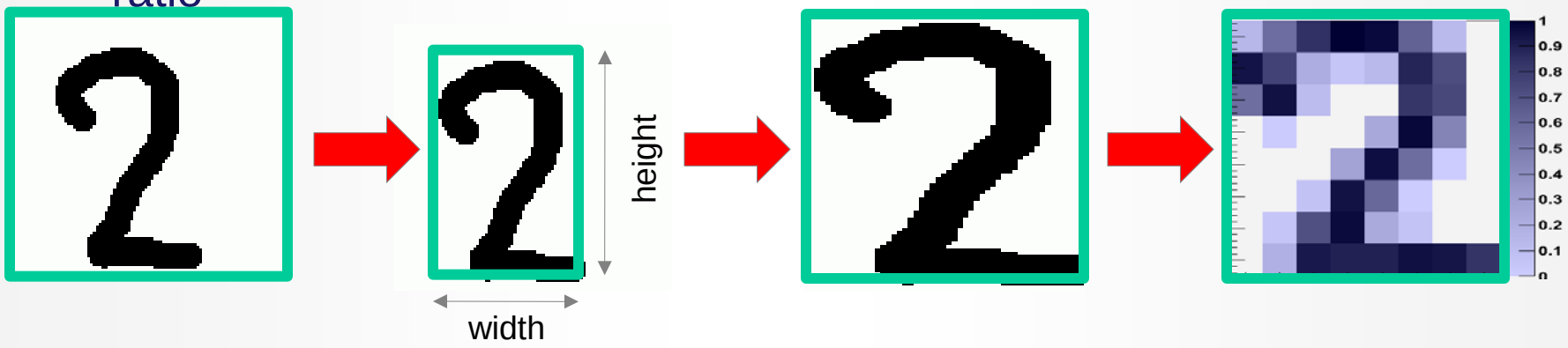
- Input values: brightness of each individual pixel

  → need to reduce number of pixels

  → preprocessing necessary

Handwritten digits/letters

- Preprocessing:
  - [step 1] Find frame around digit, determine aspect ratio
  - [step 2] Transform to aspect ratio=1
  - [step 3] Merge pixels into 8x8 array
  - Input to multivariate analysis: 64 pixels plus the original aspect ratio



Original digit
~100 dpi

Step (1)

Step (2)
Aspect ratio=1

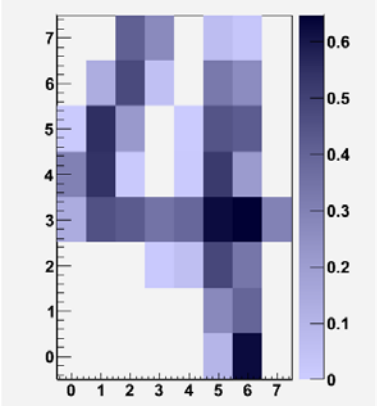
Step (3)
8x8 pixel array

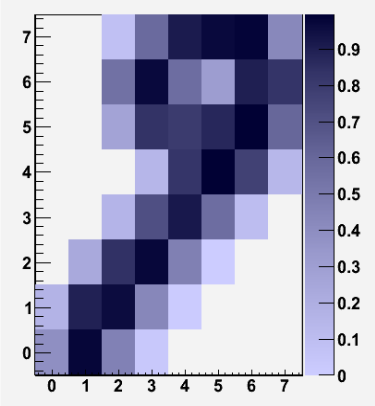
$$\text{Aspect ratio} = \frac{\text{height}}{\text{width}}$$

Pattern Recognition of handwritten digits

- Boosted Decision Trees with gradient boost (3000 trees)

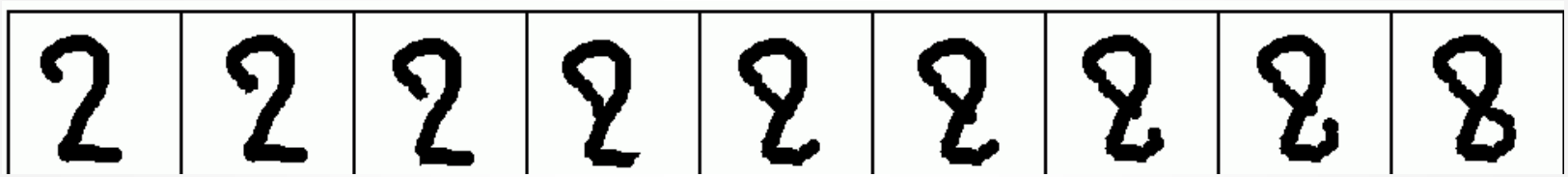
- Training: One digit is signal, all others are background
- Data sample:
  - MNIST database: 60k training digits, 10k test
  - (http://yann.lecun.com/exdb/mnist/)
  - Strict separation of test and training sample
  - persons contributing to training sample do NOT contribute to test sample (and vice versa).

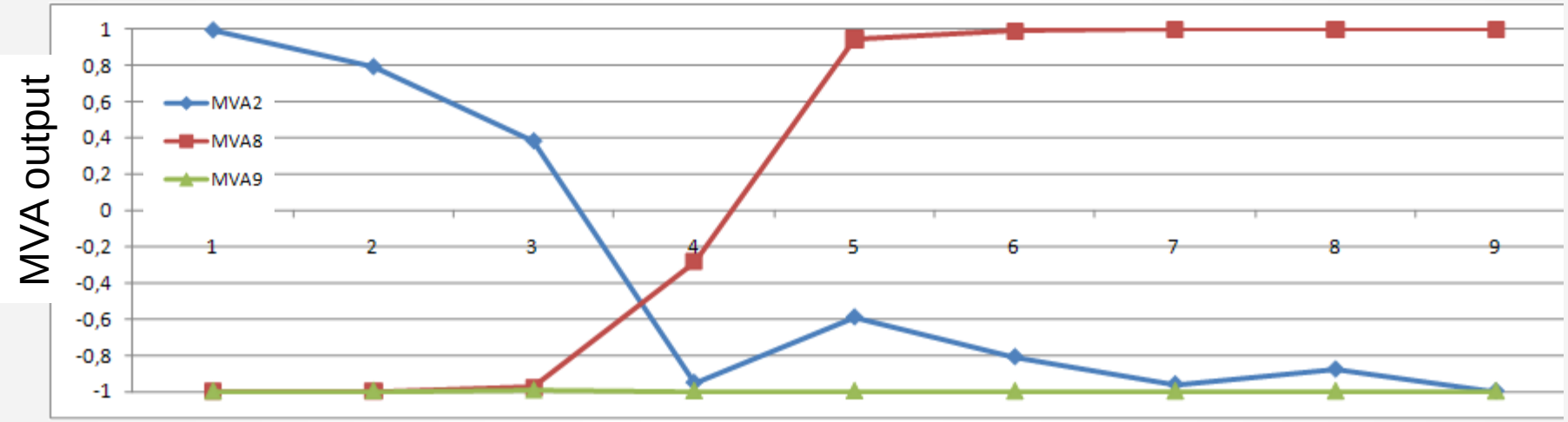




Pattern Recognition of handwritten digits

Example: stepwise morphing of “2“ into “8“





Selected as

“2“ “2“ “2“ “8“ “8“ “8“ “8“ “8“ “8“

Output digit determined by MVA with largest output value

Credits

- ***T*MVA is open source software**
- **Use & redistribution of source permitted according to terms in BSD license**
- **Several similar data mining efforts with rising importance in most fields of science and industry**

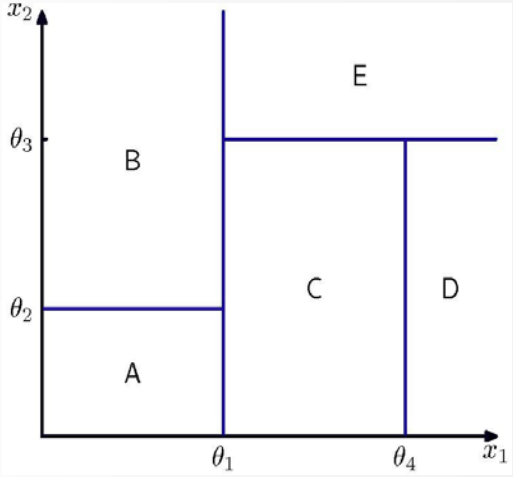
Contributed to TMVA have: Andreas Hoecker (CERN, Switzerland), Jörg Stelzer (CERN, Switzerland), Peter Speckmayer (CERN, Switzerland), Jan Therhaag (Universität Bonn, Germany), Eckhard von Toerne (Universität Bonn, Germany), Helge Voss (MPI für Kernphysik Heidelberg, Germany), Moritz Backes (Geneva University, Switzerland), Tancredi Carli (CERN, Switzerland), Asen Christov (Universität Freiburg, Germany), Or Cohen (CERN, Switzerland and Weizmann, Israel), Krzysztof Danielowski (IFJ and AGH/UJ, Krakow, Poland), Dominik Dannheim (CERN, Switzerland), Sophie Henrot-Versille (LAL Orsay, France), Matthew Jachowski (Stanford University, USA), Kamil Kraszewski (IFJ and AGH/UJ, Krakow, Poland), Attila Krasznahorkay Jr. (CERN, Switzerland, and Manchester U., UK), Maciej Kruk (IFJ and AGH/UJ, Krakow, Poland), Yair Mahalalel (Tel Aviv University, Israel), Rustem Ospanov (University of Texas, USA), Xavier Prudent (LAPP Annecy, France), Arnaud Robert (LPNHE Paris, France), Christoph Rosemann (DESY), Doug Schouten (S. Fraser U., Canada), Fredrik Tegenfeldt (Iowa University, USA, until Aug 2007), Alexander Voigt (CERN, Switzerland), Kai Voss (University of Victoria, Canada), Marcin Wolter (IFJ PAN Krakow, Poland), Andrzej Zemla (IFJ PAN Krakow, Poland), Jiahang Zhong (Academica Sinica, Taipeh).

BACKUP

test

Boosted Decision Trees

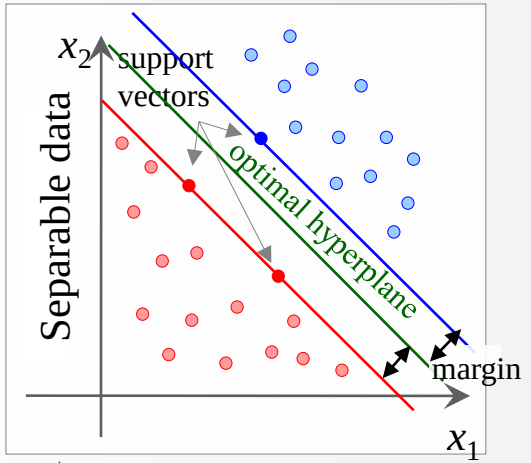
- Grow a forest of decision tree and determine the event class/target by majority vote
- Weights of misclassified events are increased for the next iteration
- Advantages:
  - ignores weak variables
  - works out of the box
- Disadvantages:
  - vulnerable to overtraining

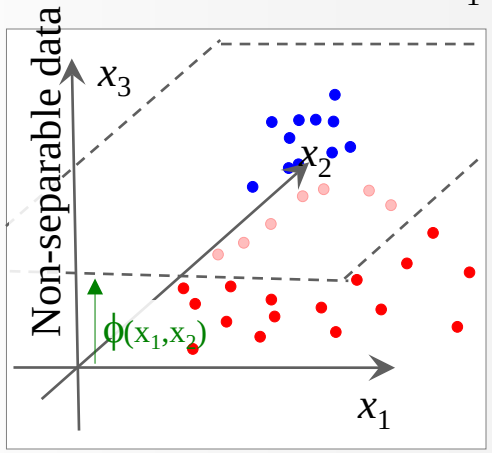


Performance		Speed		Robustness		Curse of Dim.	Transparency	Regression	
No/linear correlations	Nonlinear correlations	Training	Response	Overtraining	Weak input vars			1D	multi D
☺	☺	😐	😐	😐/☹	☺	☺	😐/☹	☺	☹

Support Vector Machines

- Linearly separable data:
  - find separating hyperplane (Fisher)
- Non-separable data:
  - fransform variables into a high dimensional space where linear separation is possible
- Advantages:
  - flexibility combined with robustness
- Disadvantages:
  - slow training
  - complex algorithm





1

Performance		Speed		Robustness		Curse of Dim.	Transparency	Regression	
No/linear correlations	Nonlinear correlations	Training	Response	Overtraining	Weak input vars			1D	multi D
☺	☺	☹	😐	😐	😐	😐	☹	☺	☺