TMVA
A Toolkit for MultiVariate Data Analysis with ROOT

Andreas Höcker (ATLAS), Helge Voss (LHCb), Fredrik Tegenfeld (ATLAS),

Kai Voss (ex. ATLAS), Joerg Stelzer (ATLAS)

supply an environment to easily:

- **apply a (large) variety of sophisticated data selection algorithms**
- **have them all trained and tested**
- **choose the best one for your selection problem**

http://tmva.sourceforge.net/

Motivation/Outline

Idea: Rather than just implementing new MVA techniques and making them available in ROOT in the way TMulitLayerPerceton does:

- have one common interface to different MVA methods
  - →easy to use
  - →easy to compare many different MVA methods
- train/test on same data sample
- have one common place for possible pre-processing (decorrelation of variables) available for all MVA selection algorithms

Outline:

- Introduction: MVAs, what / where / why
- the MVA methods available in *T*MVA
- demonstration with toy examples
- Summary/Outlook

Introduction to MVA

- At the beginning of each physics analysis:
  - select your event sample, discriminating against background
  - b-tagging
  - ...
- Or even earlier:
  - e.g. particle identification, pattern recognition (ring finding) in RICH detectors
  - trigger applications
  - discriminate “tau-jets” from quark-jets

➔ Always one uses several variables in some sort of combination:

- MVA -- MulitVariate Analysis:
  - nice name, means nothing else but:

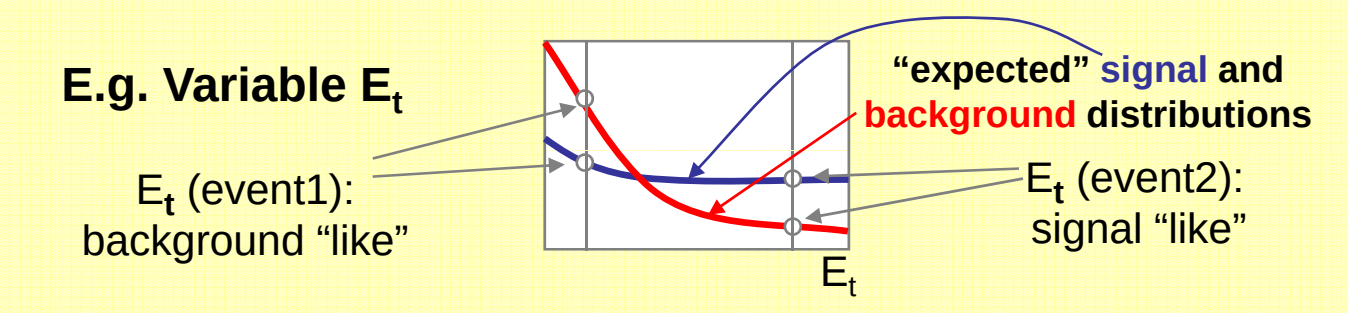
Use several observables from your events to form ONE combined variable and use this in order to discriminate between “signal” and “background”

Introduction to MVAs

- **sequence of cuts ←→ multivariate methods**
  - **sequence of cuts is easy to understand offers easy interpretation**
  - **sequences of cuts are often inefficient!! e.g. events might be rejected because of just ONE variable, while the others look very "signal like"**

MVA: several observables ➔ One selection criterium

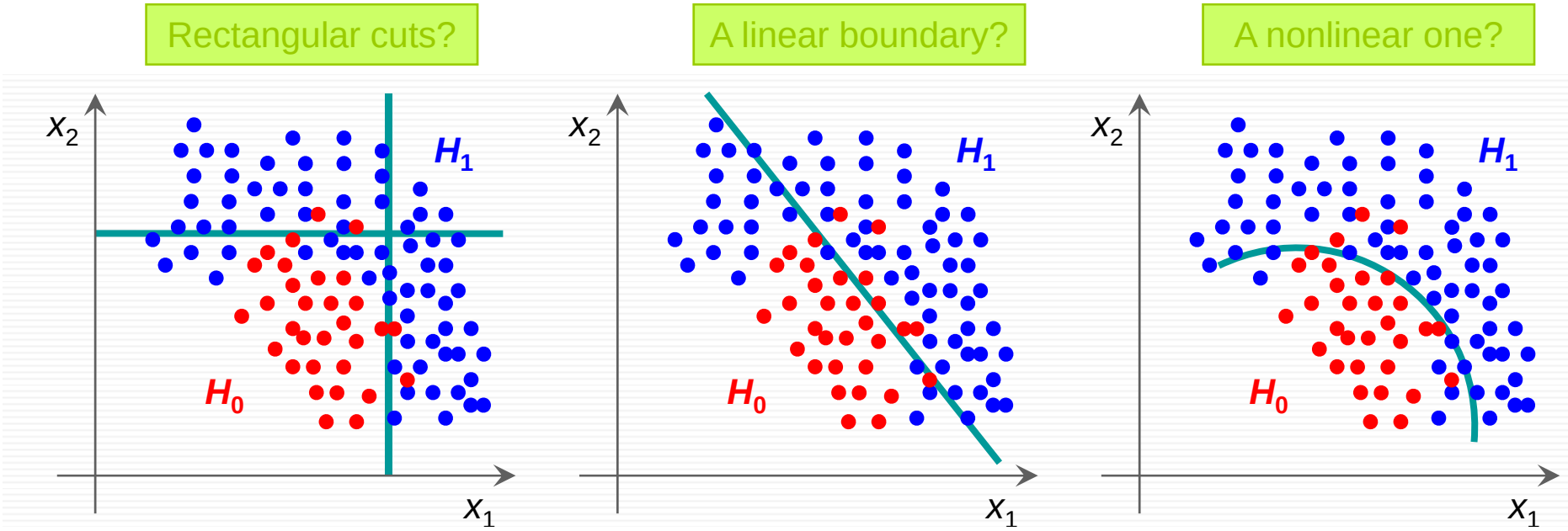
- **e.g: Likelihood selection**
  - **calculate for each observable in an event a probability that its value belongs to a "signal" or a "background" event using reference distributions (PDFs) for signal and background.**



  - **Then cut on the combination of all these probabilities**

Introduction: Event Classification

- How to exploit the information present in the discriminating variables:
- Often, a lot of information is also given in by the correlation of the variables



- Different techniques use different ways trying to exploit (all) features

  → compare and choose
- How to make a selection → let the machine learn (training)

What is *T*MVA

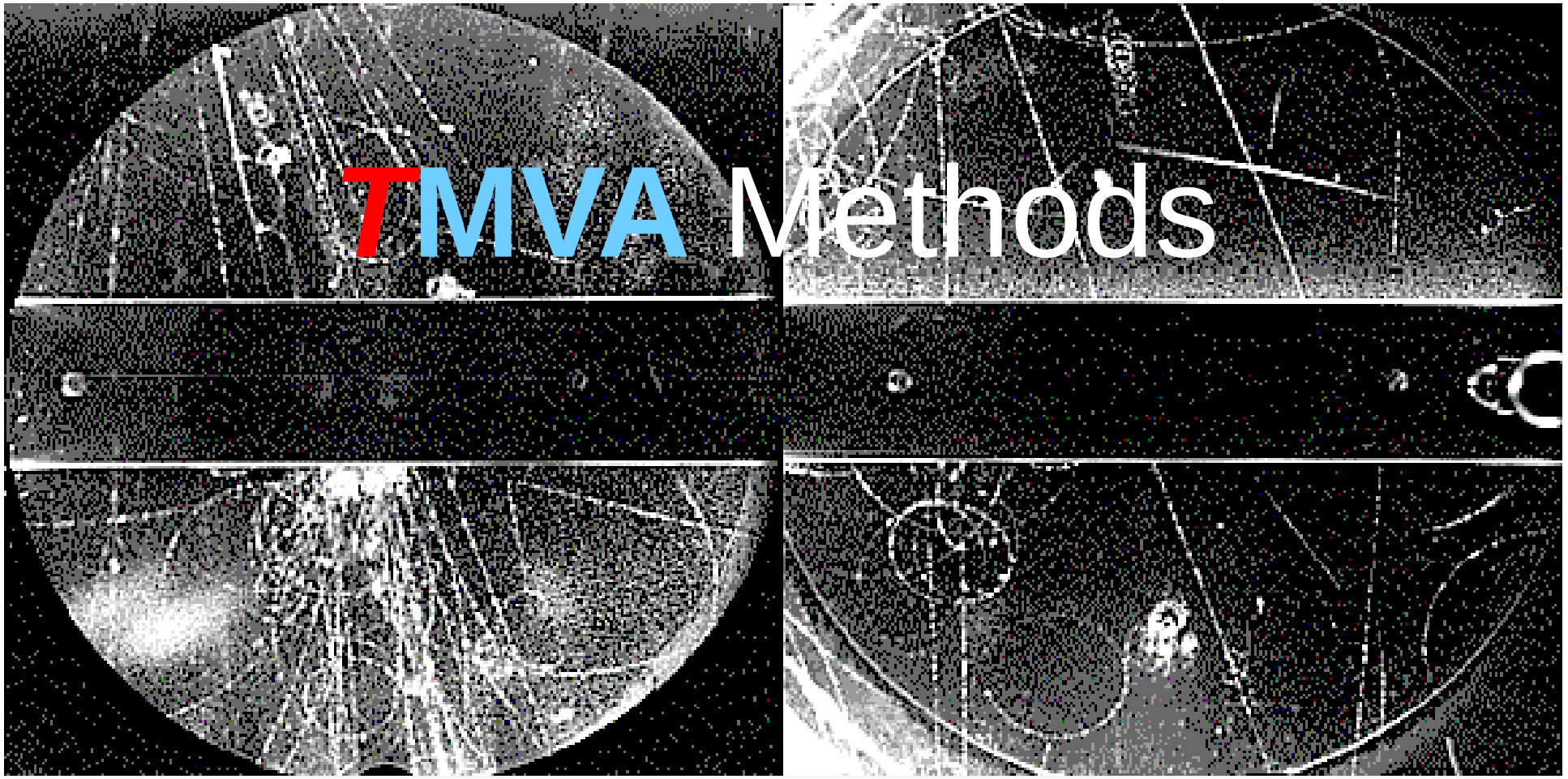
- Toolkit for Multivariate Analysis (*T*MVA) with ROOT
  - 'parallel' processing of various MVA techniques to discriminate *signal* from *background* samples.
  - ➔ easy comparison of different MVA techniques – choose the best one
- *T*MVA presently includes:
  - Rectangular cut optimisation
  - Projective and Multi-dimensional likelihood estimator
  - Fisher discriminant and H-Matrix (χ^2 estimator)
  - Artificial Neural Network (3 different implementations)
  - Boosted/bagged Decision Trees
  - Rule Fitting,
  - upcomming: Support Vector Machines, "Committee" methods
  - common pre-processing of input data: de-correlation, principal component analysis
- *T*MVA package provides training, testing and evaluation of the MVAs
- Each MVA method provides a ranking of the input variables
- MVAs produce weight files that are read by reader class for MVA application

MVA Experience

- MVAs certainly made it's way through HEP but "simple "cuts" are also still widely used

- MVAs are tedious to implement. Ready made tools often are just for one method only
  - few true comparisons between different methods are made

- Ease of use will hopefully also help to remove remaining "black box" mystic once one gains more experience in how the methods behave
  - black boxes ! how to interpret the selection ?
  - what if the training samples incorrectly describe the data ?
  - how can one evaluate systematics ?

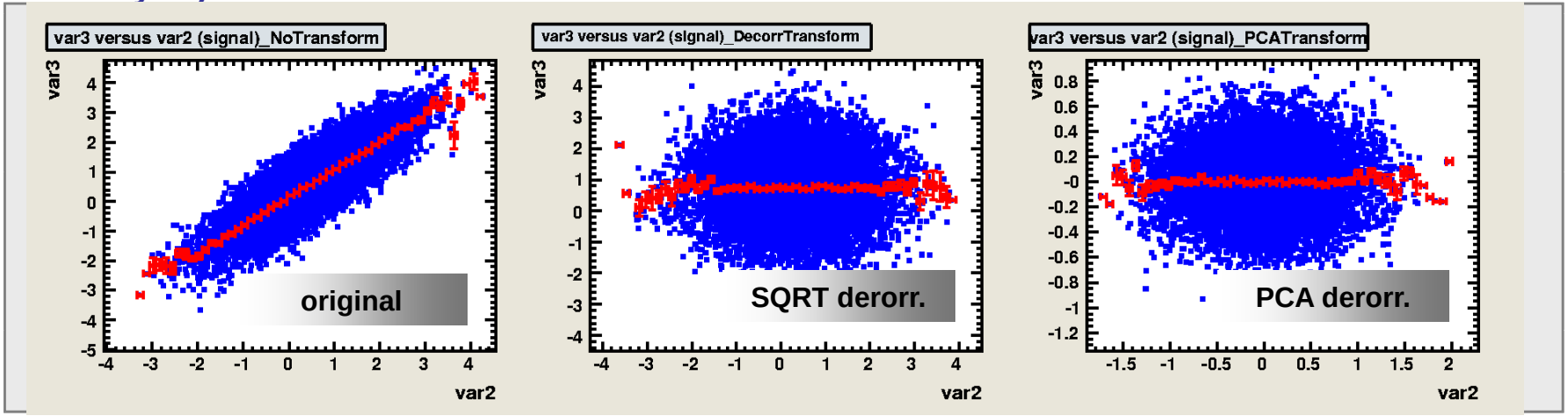

TMVA Methods

Preprocessing the Input Variables: Decorrelation

- Commonly realised for all methods in *T*MVA (centrally in DataSet class):
- Removal of linear correlations by rotating variables

- Determine *square-root C′* of correlation matrix *C*, *i.e.*, $C = C'C'$
  - compute *C′* by diagonalising *C*: $D = S^T C S \Rightarrow C' = S\sqrt{D}S^T$
  - transformation from original (x) in de-correlated variable space (x′) by: $x' = C'^{-1}x$

- Various ways to choose diagonalisation (also implemented: principal component analysis)



Cut Optimisation

- **Simplest method: cut in rectangular volume using N_{var} input variables**

$$x_{\text{cut},i_{\text{event}}} \in (0,1) \;=\; \bigcap_{v \in \{\text{variables}\}} \left\{ x_{v,i_{\text{event}}} \subset \left[x_{v,\min}, x_{v,\max} \right] \right\}$$

- **Usually training files in *T*MVA do not contain *realistic* signal and background abundance → cannot optimize for best significance ($S/\sqrt{(S+B)}$)**
  - **scan in signal efficiency [0 →1] and maximise background rejection**

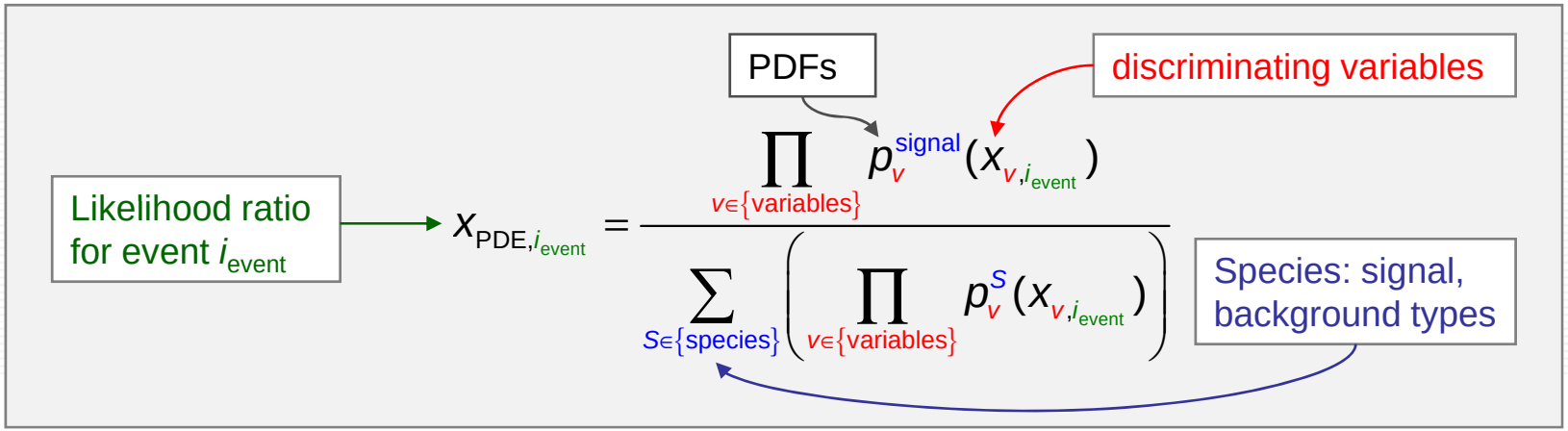
- **Technical problem: how to perform optimisation**
  - **random sampling: robust (if not too many observables used) but suboptimal**
  - **new techniques → *Genetics Algorithm* and *Simulated Annealing***

  Huge speed improvement by sorting training events in Binary Search Tree (for 4 variables we gained a factor 41)

- **do this in normal variable space or de-correlated variable space**

Projected Likelihood Estimator (PDE Appr.)

- **Combine probability for an event to be signal / background from individual variables to**



- **Assumes uncorrelated input variables**
  - **in that case it is the optimal MVA approach, since it contains *all* the information**
  - **usually it is *not true* → development of different methods**

- **Technical problem: how to implement reference PDFs**
  - **3 ways: function fitting, counting, parametric fitting (splines, kernel est.)**

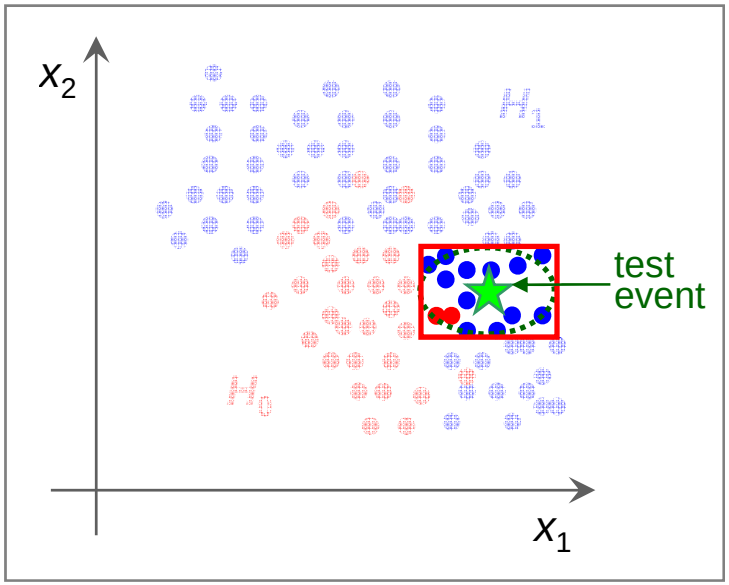
difficult to automate

automatic,unbiased, but suboptimal

easy to automate, can create artefacts
TMVA uses: Splines0-5, Kernel estimators

Multidimensional Likelihood Estimator

- Generalisation of 1D PDE approach to N_{var} dimensions
- Optimal method – in theory – if "true N-dim PDF" were known
- Practical challenges:
  - derive N-dim PDF from training sample
- *T*MVA implementation:
  - count number of signal and background events in "vicinity" of a data event → fixed size or adaptive
  - volumes can be rectangular or spherical
  - use multi-D kernels (Gaussian, triangular, …) to weight events within a volume
  - speed up range search by sorting training events in Binary Trees



Carli-Koblitz, NIM A501, 576 (2003)

Fisher Discriminant (and H-Matrix)

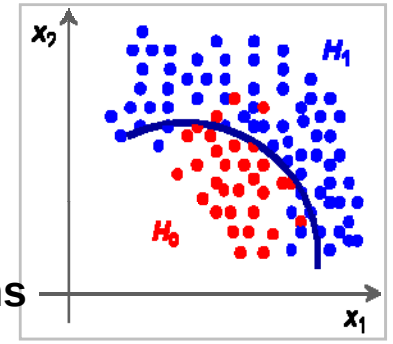
- **Well-known, simple and elegant MVA method:**
  - **determine linear boundary between signal and background in transformed variable space where:**
    - **linear correlations are removed**
    - **mean values of signal and background are "pushed" as far apart as possible**
- **optimal for linearly correlated Gaussians with equal RMS' and different means**
  - **no separation if equal means and different RMS (shapes)**
- **Computation of the trained Fisher MVA couldn't be simpler:**

$$x_{\text{Fisher},i_{\text{event}}} \propto \sum_{v \in \{\text{variables}\}} \left\{ x_{v,i_{\text{event}}} \cdot F_v \right\}$$

"Fisher coefficients"

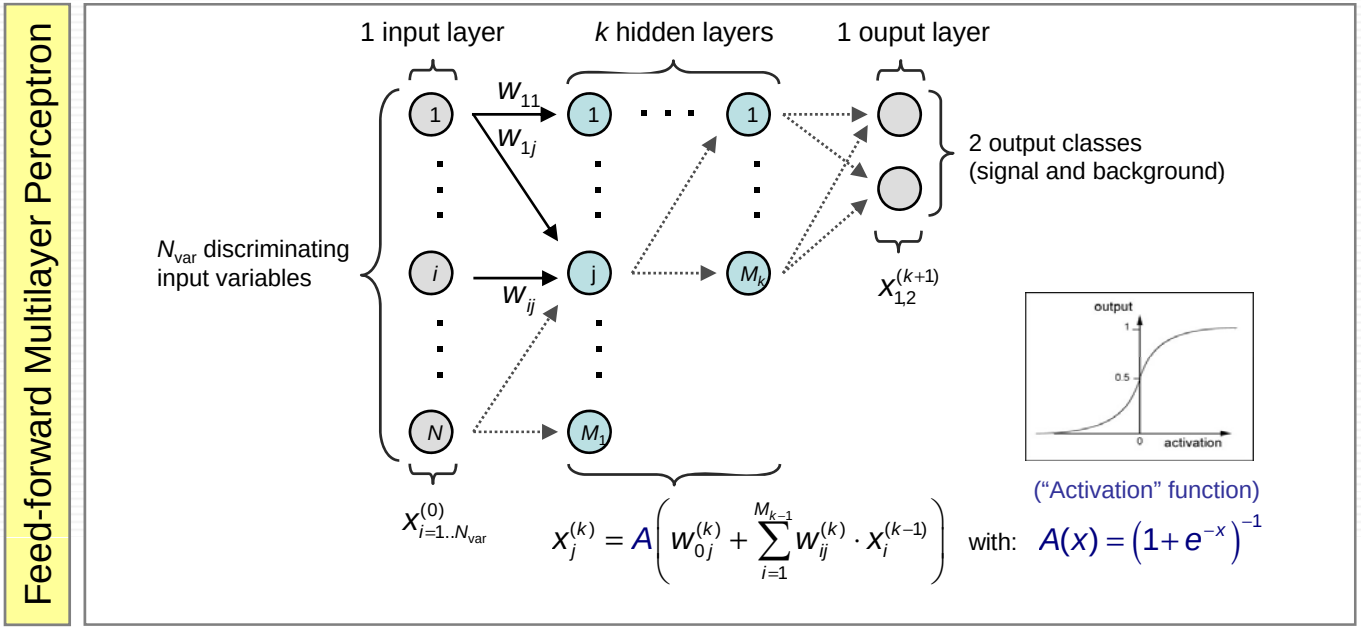
Artificial Neural Network (ANN)

- Get a non-linear classifier response by "activating" output nodes using non-lieaear weights (activation)



- Nodes are called "neurons" and arranged in series

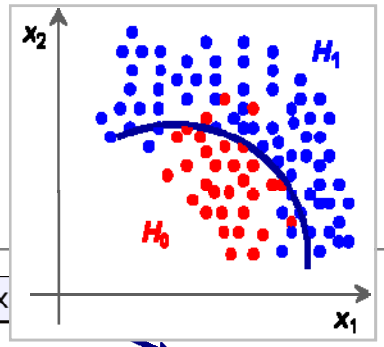
→ **Feed-Forward Multilayer Perceptrons (3 different implementations in TMVA)**



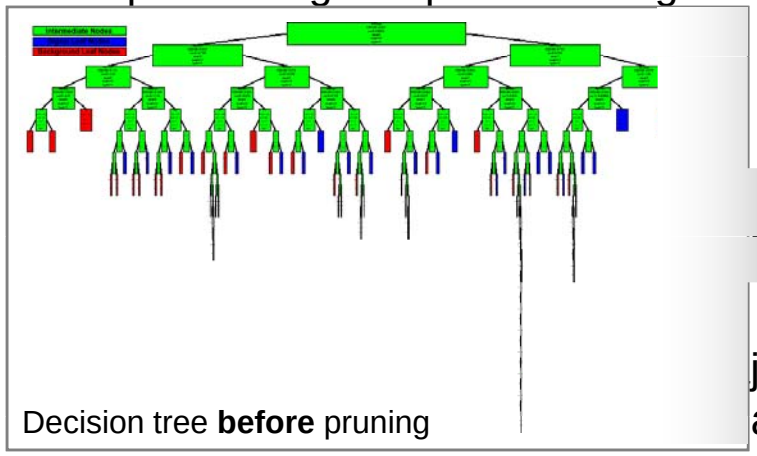
- Taining→ adjust weight of each input variable to node-activation using training events

Decision Trees

- **sequential application of “cuts” which splits the data into nodes, and the final nodes (leaf) classifies an event as signal or background**



- **Training: growing a decision tree**
  - Start with Root node
  - Split training sample according to


Decision tree **before** pruning

Decision tree **after** pruning

jority of e
assified a

- **Bottom up Pruning:**
  - **remove statistically insignificant nodes (avoid overtraining)**

Boosted Decision Trees

- **Decision Trees:** used since a long time in general "data-mining" applications, less known in HEP (although very similar to "simple Cuts")
- **Advantages:**
  - easy to interpret: visualisation in a 2D tree
  - independent of monotone variable transformation: immune against outliers
  - usless/weak variables are ignored
- **Disadvatages:**
  - instability: small changes in training sample can give large changes in tree structure
- **Boosted Decision Trees (1996): combining several decision trees (forest) derived from one training sample via the application of event weights into ONE mulitvariate event classifier by performing "majority vote"**
  - *e.g.* AdaBoost: wrong classified training events are given a larger weight
  - bagging, random weights → re-sampling with replacement
  - bagging/boosting: means of creating basis functions: final classifier is a linear combination of those

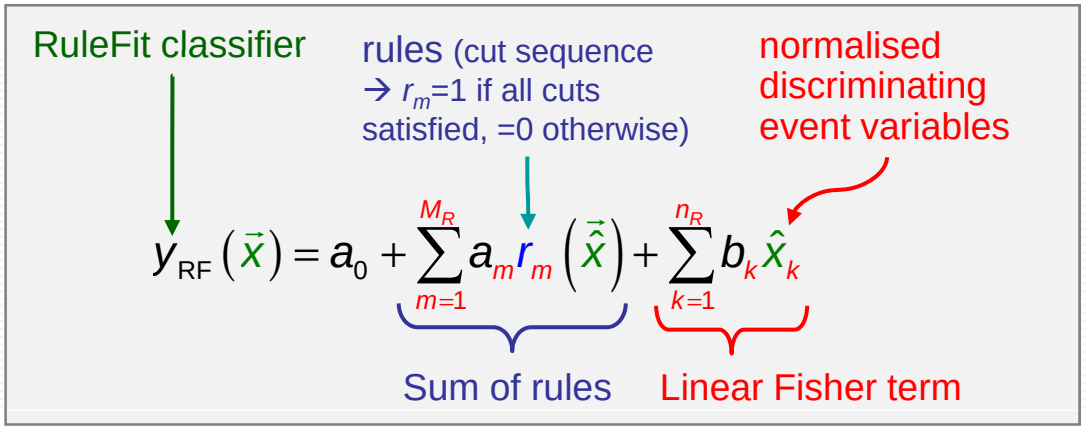
Rule Fitting

(Predictive Learing via Rule Ensembles)

- **Following *RuleFit* from Friedman-Popescu:**

Friedman-Popescu, Tech Rep, Stat. Dpt, Stanford U., 2003

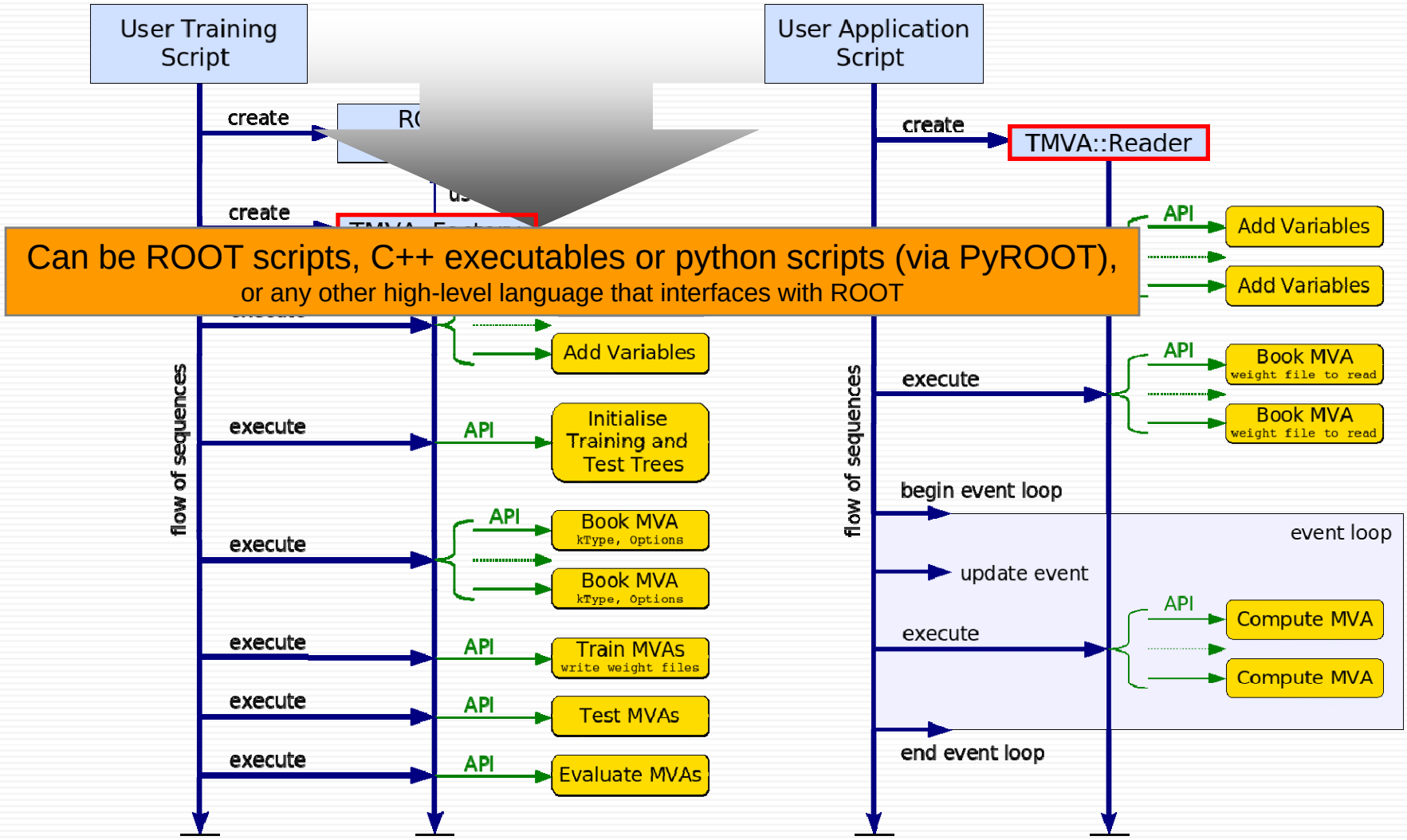
- **Model is a linear combination of *rules*; a *rule* is a *sequence of cuts:***



- **The problem to solve is:**
  1. **create the rule ensemble → created from a set of decision trees**
  2. **fit the coefficients → "Gradient directed regularization" (Friedman et al)**

- **Fast, robust and good performance**

Using TMVA in Training and Application



A Complete Example Analysis

```
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");

   TTree *signal     = (TTree*)input->Get("TreeS");
   TTree *background = (TTree*)input->Get("TreeB");
   factory->AddSignalTree    ( signal,     1. );
   factory->AddBackgroundTree( background, 1.);

   factory->AddVariable("var1+var2", 'F');
   factory->AddVariable("var1-var2", 'F');
   factory->AddVariable("var3", 'F');
   factory->AddVariable("var4", 'F');

   factory->PrepareTrainingAndTestTree("", "NSigTrain=3000:NBkgTrain=3000:SplitMode=Random:!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->TestAllMethods();
   factory->EvaluateAllMethods();

   outputFile->Close();
   delete factory;
}
```

create Factory

give training/test trees

tell which variables

select the MVA methods

train,test and evaluate

Example Application

```
void TMVApplication(  )
{
   TMVA::Reader *reader = new TMVA::Reader("!Color");

   Float_t var1, var2, var3, var4;
   reader->AddVariable( "var1+var2", &var1 );
   reader->AddVariable( "var1-var2", &var2 );
   reader->AddVariable( "var3", &var3 );
   reader->AddVariable( "var4", &var4 );

   reader->BookMVA( "MLP method",  "weights/MVAnalysis_MLP.weights.txt" );

   TFile *input = TFile::Open("tmva_example.root");
   TTree* theTree = (TTree*)input->Get("TreeS");

   Float_t userVar1, userVar2;
   theTree->SetBranchAddress( "var1", &userVar1 );
   theTree->SetBranchAddress( "var2", &userVar2 );
   theTree->SetBranchAddress( "var3", &var3 );
   theTree->SetBranchAddress( "var4", &var4 );

   for (Long64_t ievt=3000; ievt<theTree->GetEntries();ievt++) {
      theTree->GetEntry(ievt);

      var1 = userVar1 + userVar2;
      var2 = userVar1 - userVar2;
      cout << reader->EvaluateMVA( "MLP method"          ) <<endl;
   }

   delete reader;
}
```

create Reader

tell it about the variables

selected MVA method

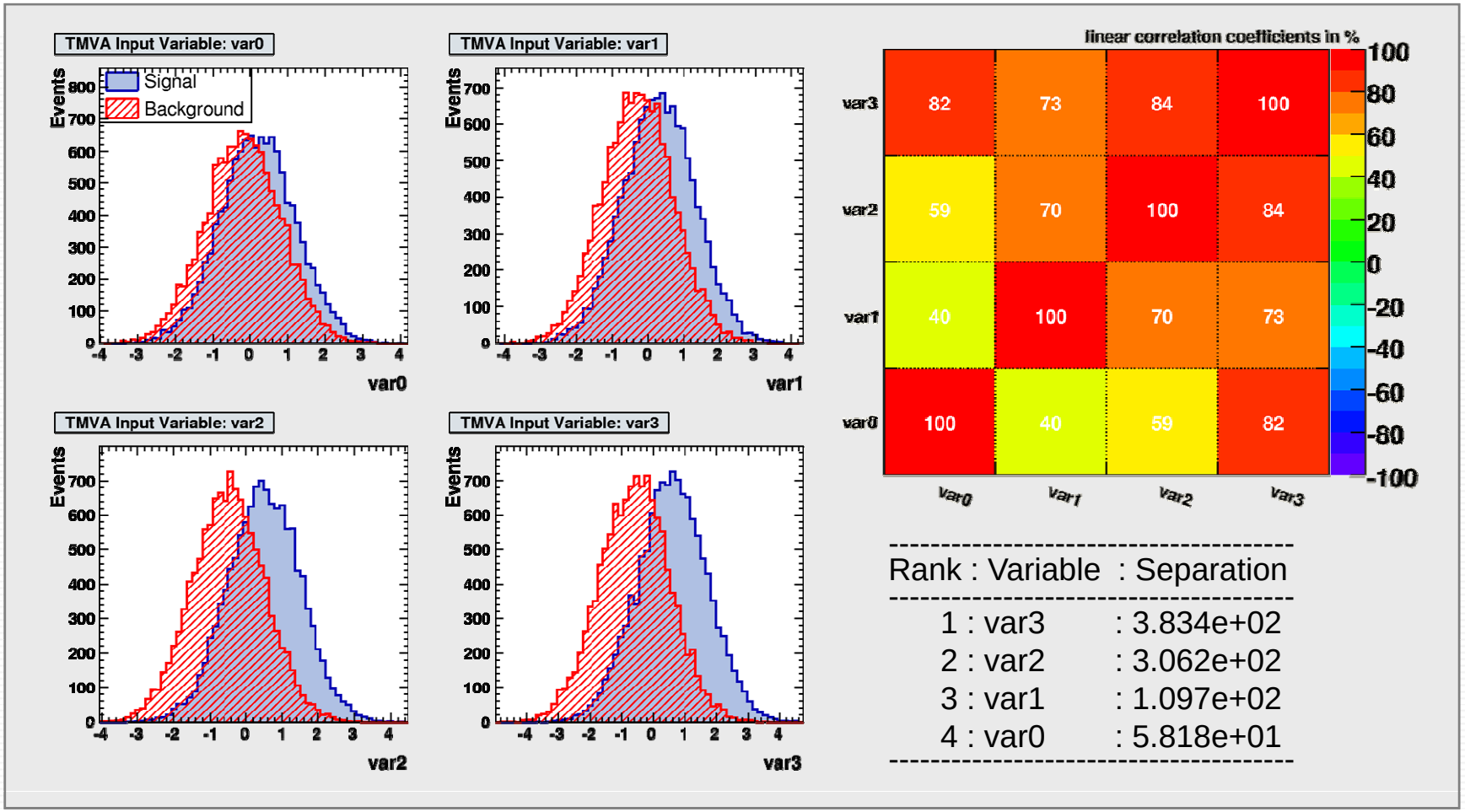
set tree variables

event loop

calculate the MVA output

A purely academic Toy example

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



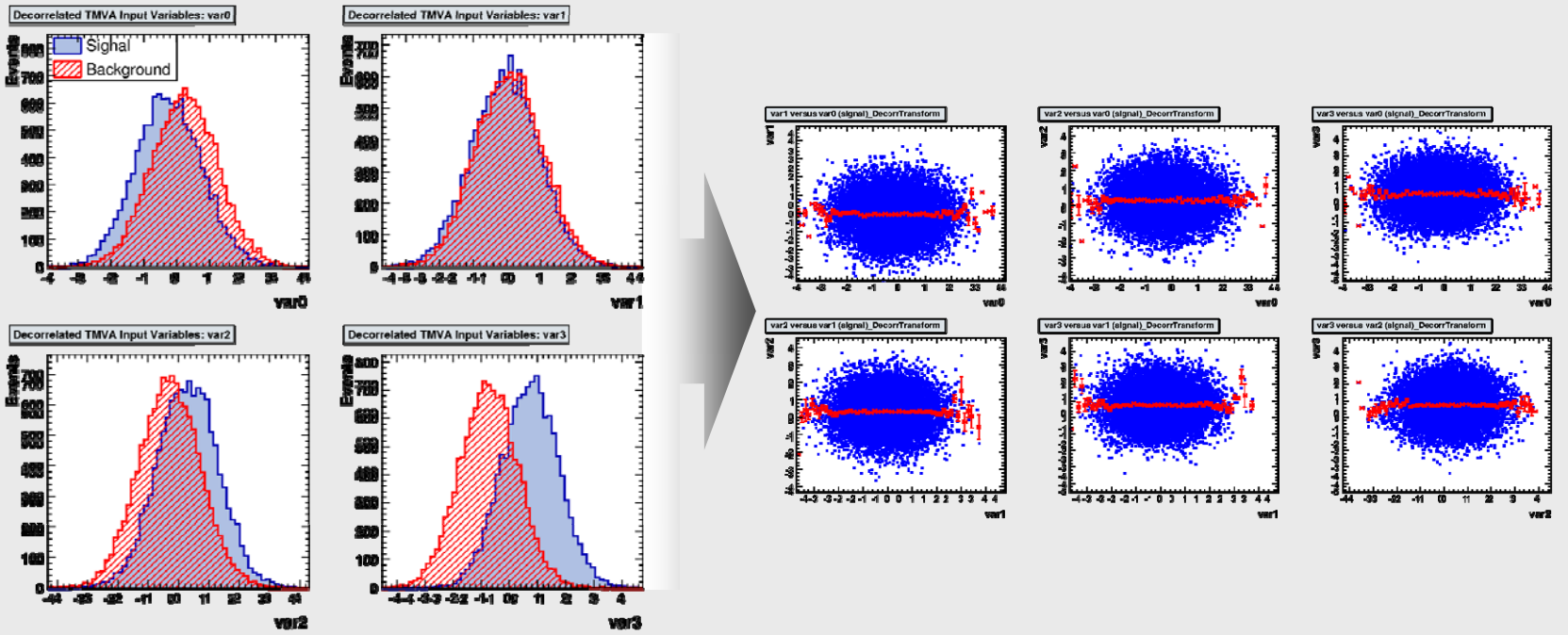
Rank : Variable : Separation

---

1 : var3 : 3.834e+02
2 : var2 : 3.062e+02
3 : var1 : 1.097e+02
4 : var0 : 5.818e+01

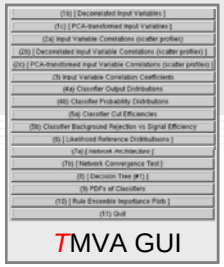
Preprocessing of input Variables

- **Decorrelation of variables before the training is usful for THIS example.**

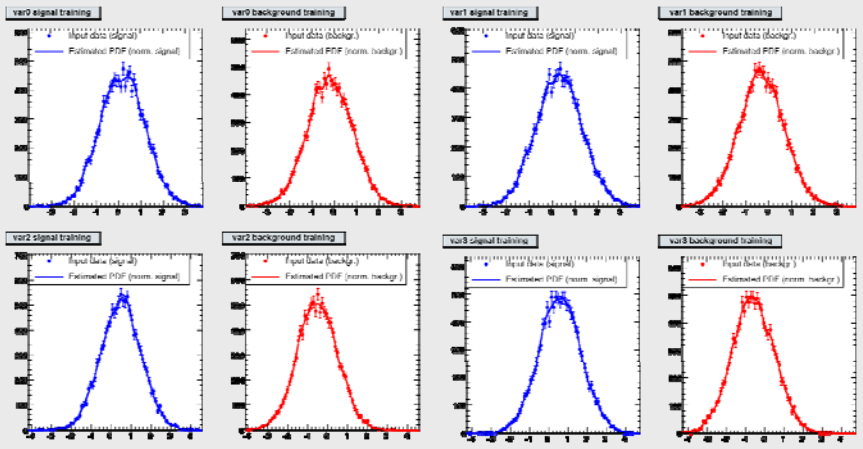


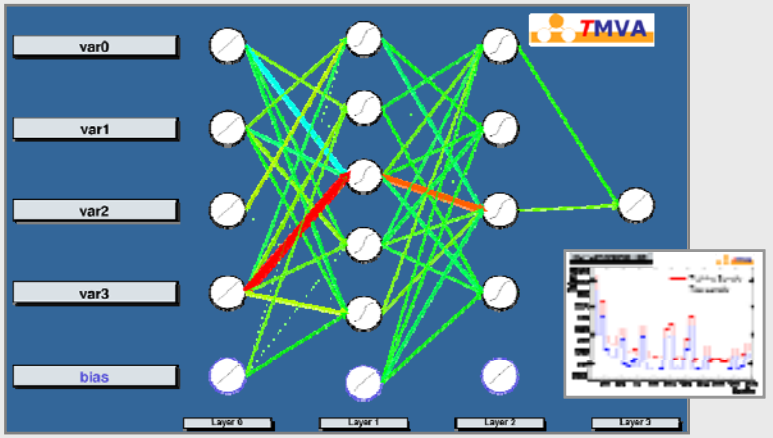
- Similar distributions for PCA
- Note that in cases with non-Gaussian distributions and/or nonlinear correlations decorrelation may do more harm than any good

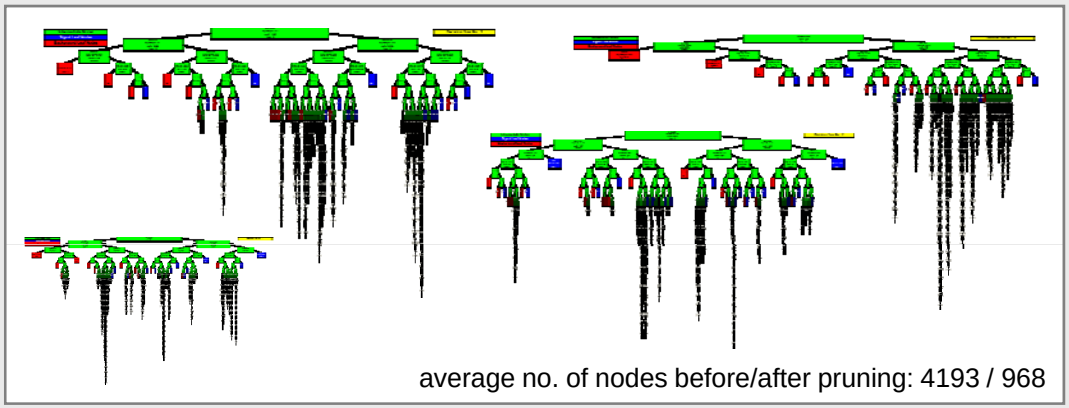
Validating the classifiers



- Projective likelihood PDFs, MLP training, BDTs, ...



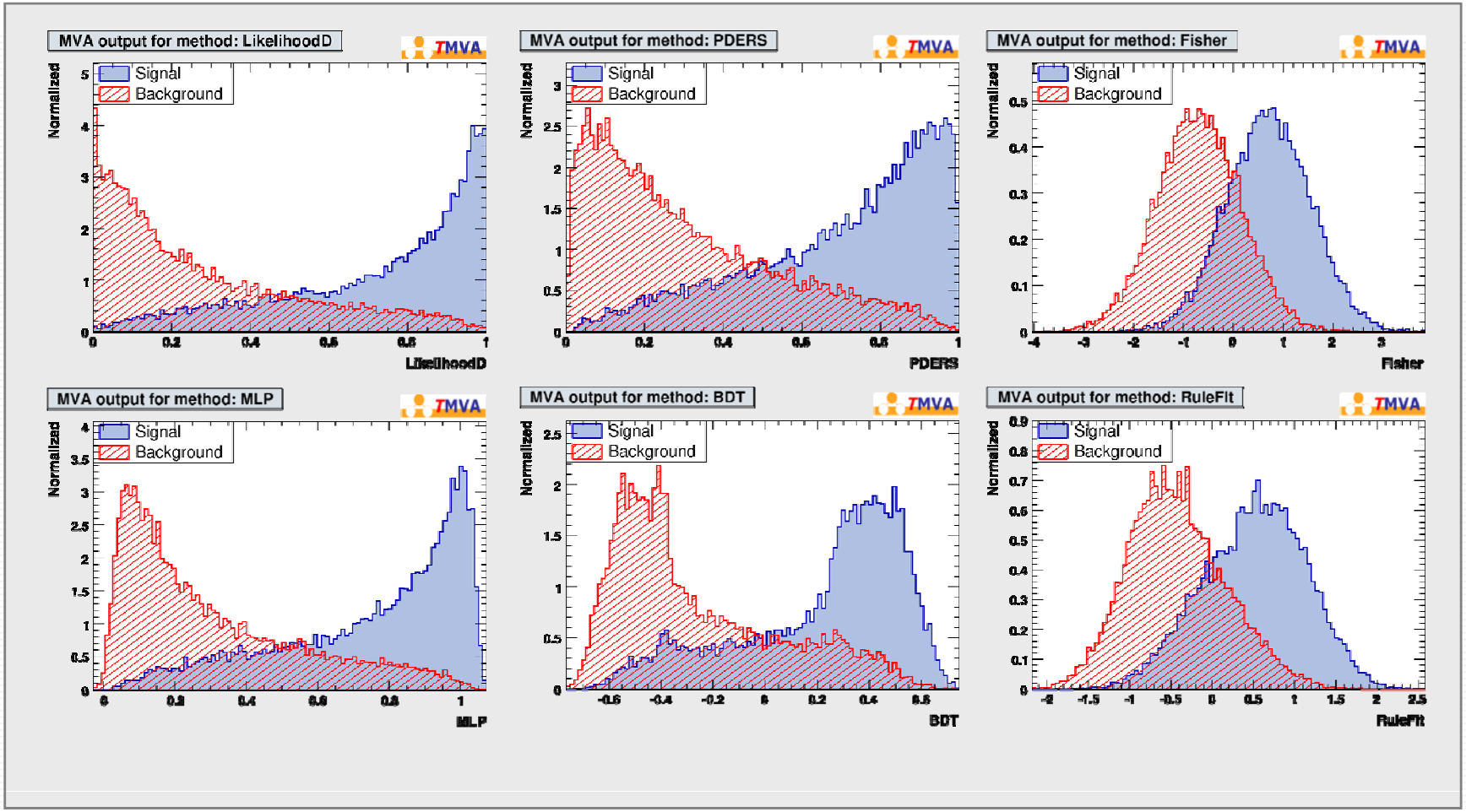




average no. of nodes before/after pruning: 4193 / 968

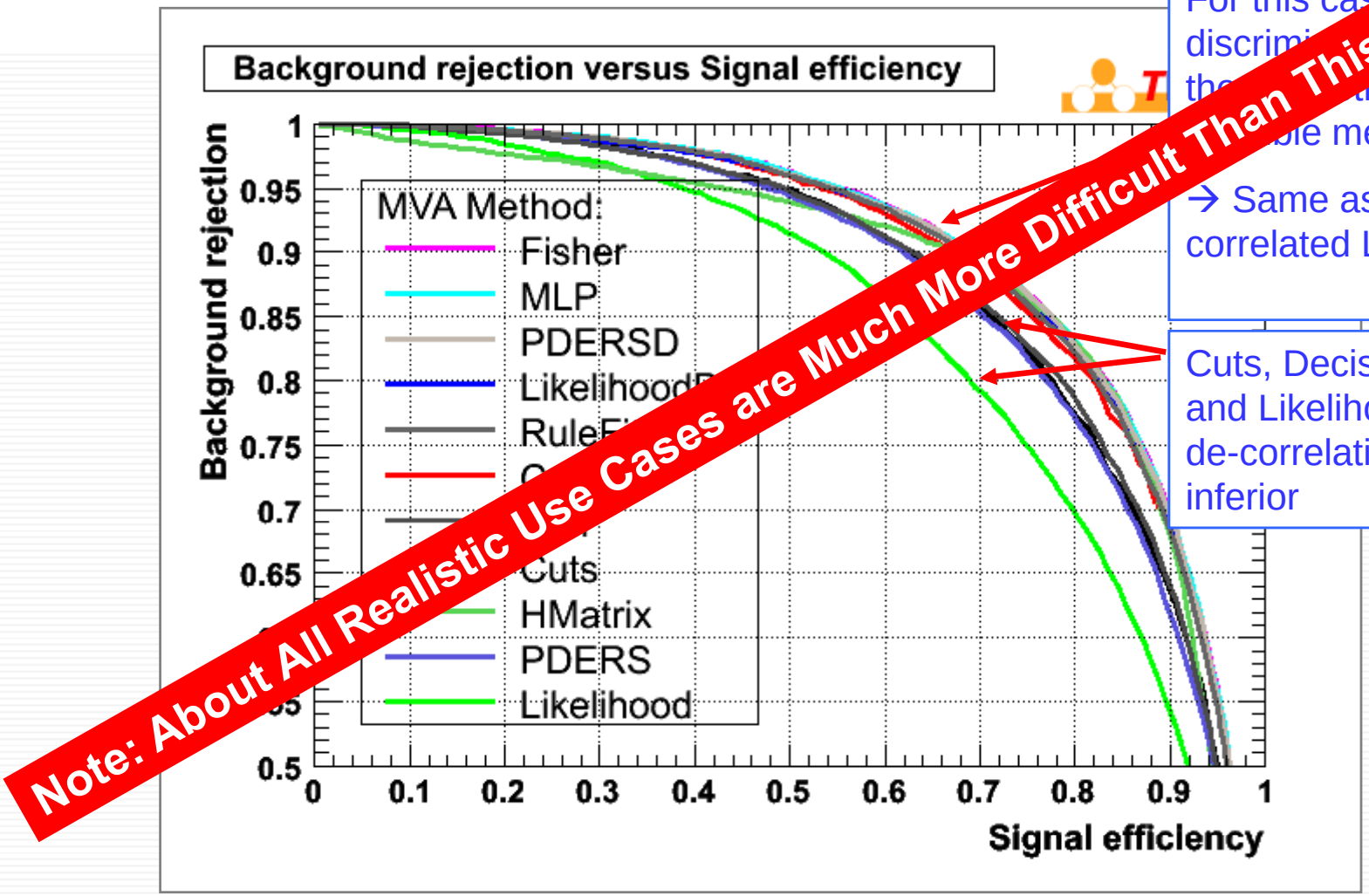
Evaluation Output

- *TMVA* output distributions for Fisher, Likelihood, BDT and MLP



Evaluation Output

- *T*MVA output distributions for Fisher, Likelihood, BDT and MLP…



Note: About All Realistic Use Cases are Much More Difficult Than This One

For this case[illegible]er discrimi[illegible]ovides the[illegible]tically 'best' [illegible]ble method

→ Same as de-correlated Likelihood

Cuts, Decision Trees and Likelihood w/o de-correlation are inferior

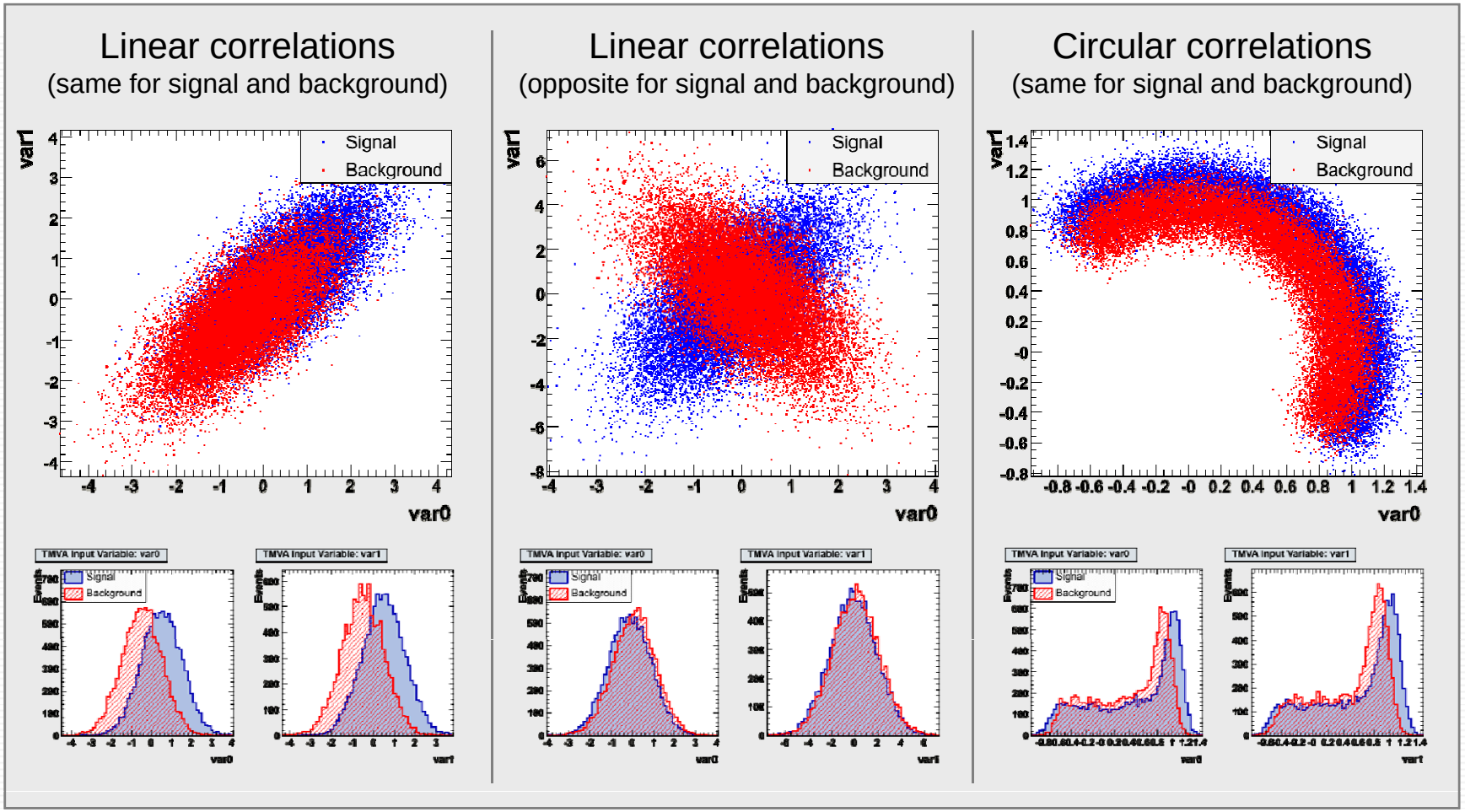
Evaluation Output (taken from *T*MVA printout)

Evaluation results ranked by best signal efficiency and purity (area)

MVA Methods:	Signal efficiency at bkg eff. (error): @B=0.01	@B=0.10	@B=0.30	Area	Separation:	Significance:
Fisher	0.268(03)	0.653(03)	0.873(02)	0.882	0.444	1.189
MLP	0.266(03)	0.656(03)	0.873(02)	0.882	0.444	1.260
LikelihoodD	0.259(03)	0.649(03)	0.871(02)	0.880	0.441	1.251
PDERS	0.223(03)	0.628(03)	0.861(02)	0.870	0.417	1.192
RuleFit	0.196(03)	0.607(03)	0.845(02)	0.859	0.390	1.092
HMatrix	0.058(01)	0.622(03)	0.868(02)	0.855	0.410	1.093
BDT	0.154(02)	0.594(04)	0.838(03)	0.852	0.380	1.099
CutsGA	0.109(02)	1.000(00)	0.717(03)	0.784	0.000	0.000
Likelihood	0.086(02)	0.387(03)	0.677(03)	0.757	0.199	0.682

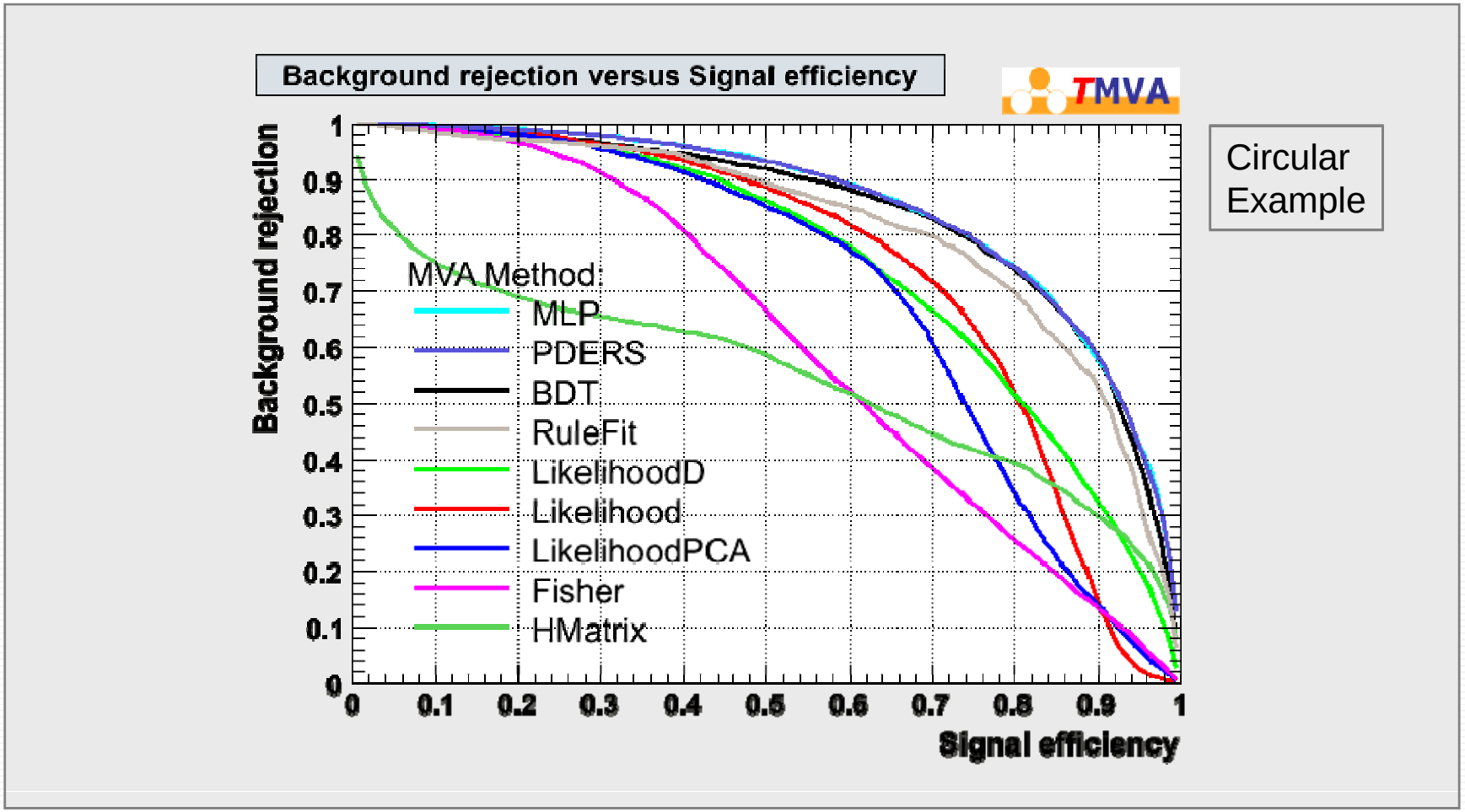
More Toys: Linear, Cross, Circular correlations

- **Illustrate the behaviour of linear and nonlinear classifiers**



Final Classifier Performance

- **Background rejection versus signal efficiency curve:**



*T*MVA Technicalities

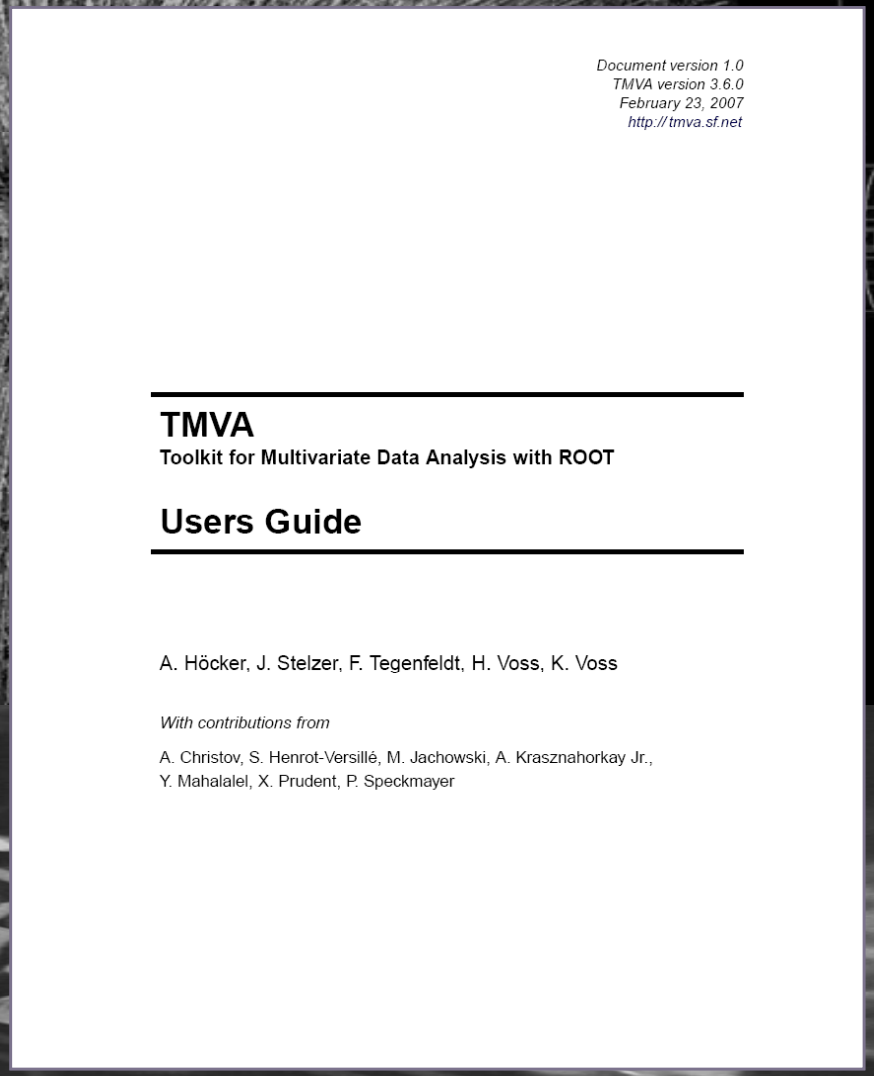
- *T*MVA releases:
  - part of the ROOT package (since Development release 5.11/06)
  - started and still available as open source package on sourceforge
    - home page: http://tmva.sourceforge.net/
    - more frequent updates than for the ROOT version. (We are still heavily developping ☺)
    - current release number 3.6.1 from 9th March 2007
  - new developers are always welcome
    - currently 4 main developers and 24 registered contributors on sourceforge

Acknowledgments: The fast development of TMVA would not have been possible without the contribution and feedback from many developers and users to whom we are indebted. We thank in particular the CERN Summer students Matt Jachowski (Stanford) for the implementation of TMVA's new MLP neural network, and Yair Mahalalel (Tel Aviv) for a significant improvement of PDERS. We are grateful to Doug Applegate, Kregg Arms, Ren\'e Brun and the ROOT team, Tancredi Carli, Elzbieta Richter-Was, Vincent Tisserand and Marcin Wolter for helpful conversations.

advertisement

We (finally) have a Users Guide !

Available from tmva.sf.net



Document version 1.0
TMVA version 3.6.0
February 23, 2007
http://tmva.sf.net

TMVA
Toolkit for Multivariate Data Analysis with ROOT

Users Guide

A. Höcker, J. Stelzer, F. Tegenfeldt, H. Voss, K. Voss

With contributions from

A. Christov, S. Henrot-Versillé, M. Jachowski, A. Krasznahorkay Jr.,
Y. Mahalalel, X. Prudent, P. Speckmayer

*T*MVA Users Guide
68pp, incl. code examples
submitted to arXiv:physics

Concluding Remarks

- *T*MVA is still a young project!
  - first release on sourceforge March 8, 2006
  - now also as part of the ROOT package

- *T*MVA provides the training and evaluation tools, but the decision which method is the best is certainly depends on the use case → train several methods in parallel and see what is best for YOUR analysis
  - also provides set of ROOT macros to visualize the results
- Most methods can be improved over default by optimising the training options

- Aimed for easy to use tool to give access to many different "complicated" selection algorithms
- Already have a number of users, but still need more real-life experience

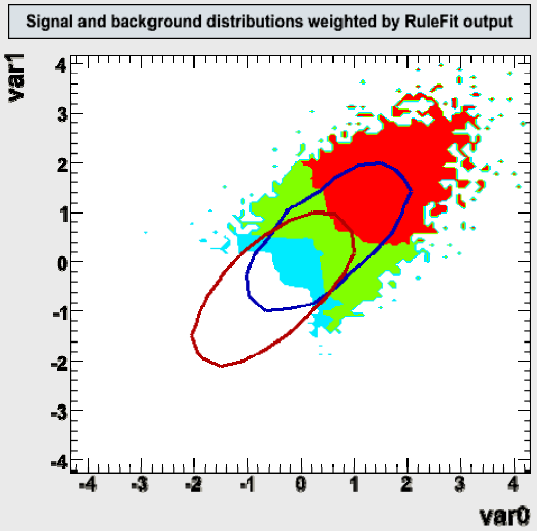
Outlook

- **We will continue to improve**
  - **the selection methods already implemented**
  - **flexibility of the data interface**
- **New Methods are under development**
  - **Support Vector Machines**
  - **Bayesian Classifiers**
  - **“Committee Method” → combination of different MVA techniques**

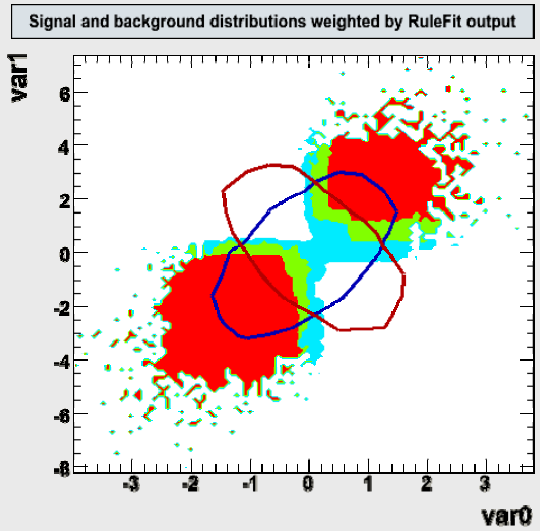
Illustustration: Events weighted by MVA-response:

- **How well do the classifier resolve the various correlation patterns ?**

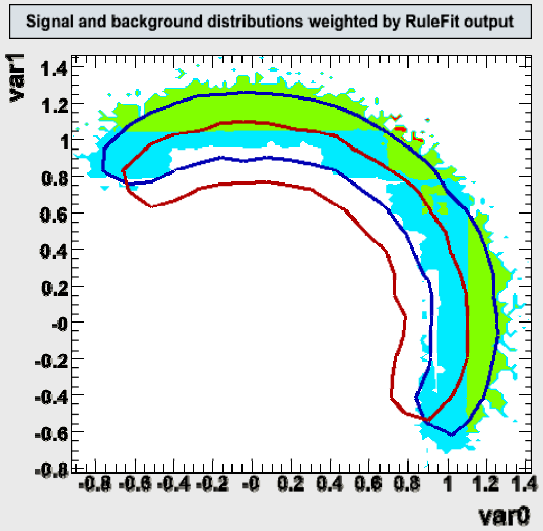
Linear correlations
(same for signal and background)



Linear correlations
(opposite for signal and background)



Circular correlations
(same for signal and background)



Stability with respect to irrelevant variables

- Toy example with 2 discriminating and 4 non-discriminating variables ?