What to expect from TMVA - The Toolkit for Multivariate Analysis in ROOT

Helge Voss (MPIK, Heidelberg)

(for TMVA A.Hoecker, E.v.Toerne, H.Voss, J.Teerhag, J.Stelzer, P.Speckmayer)

ROOT Users Workshop , Saas Fee, 11-14 March 2013

Outline

- **Introduction to TMVA**
  - Classification and Regression
  - Highlight of what is “new”-ish
- **Using TMVA**
  - example walk through
  - general remarks on MVA’s
    - do’s and don’ts
    - what about systematic errors

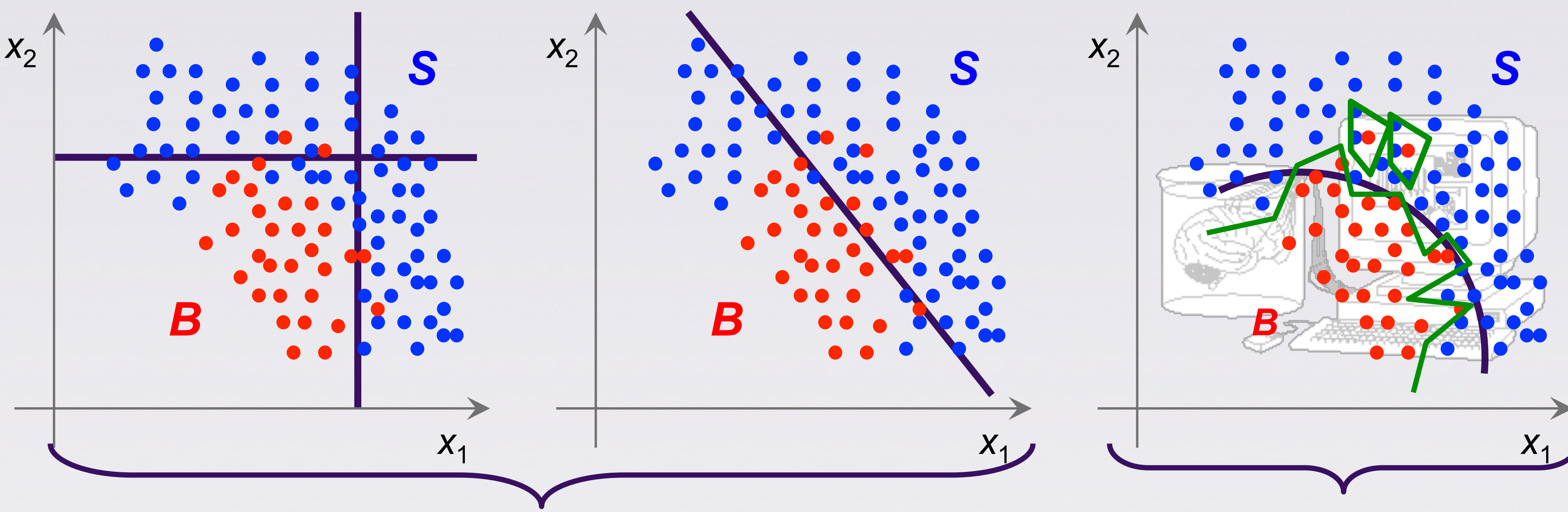
  } too bad but: → backup slides only
  - real application examples
- **Plans**
- **Summary**

Event Classification

- **Discriminate *Signal* from *Background***
  - select events of type *S* ?
  - we have discriminating variables x_1, x_2, …

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Rectangular cuts? | A linear boundary? | A nonlinear one?



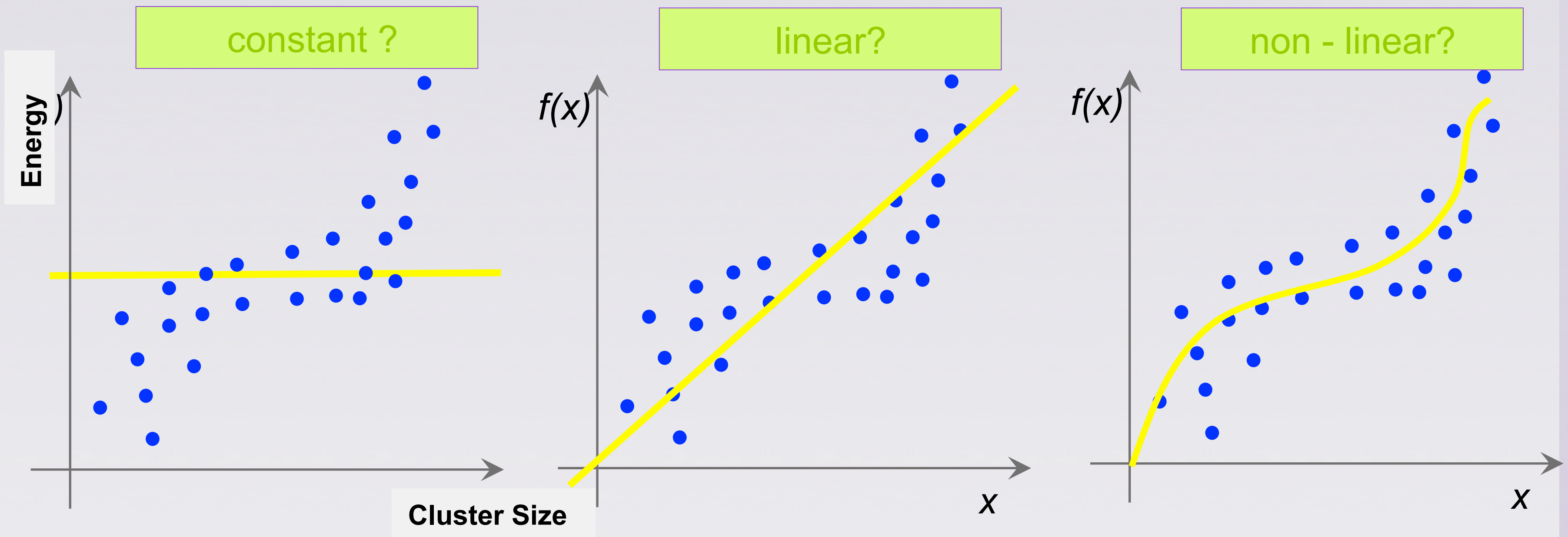
Low variance (stable), high bias methods

High variance, small bias methods

- Which model/class ? Pro and cons ?
- ***T*MVA** helps to decide on the model and finds the "optimal" boundary!

Regression

- estimate "functional behaviour" from a set of 'known measurements" ?
- e.g. : photon energy as function "D"-variables ECAL shower parameters + …



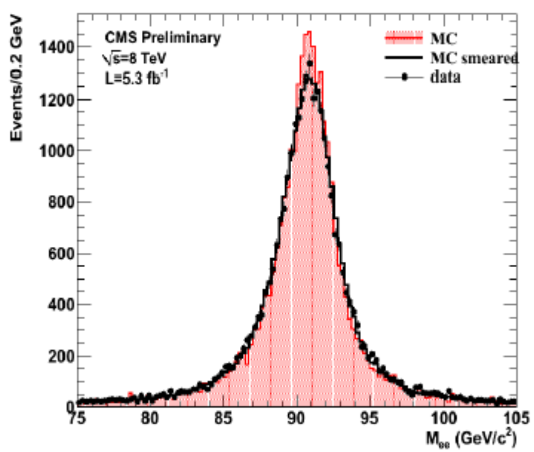
- known analytic model (i.e. n^{th} -order polynomial) → Maximum Likelihood Fit)
- no model ?
  - → "draw any kind of curve" and parameterize it?
- seems trivial ? → human brain has very good pattern recognition capabilities!
- what if you have **many** input variables? → Use ***T*MVA**

CMS Higgs Discovery
(such a nice example for MVA usage)

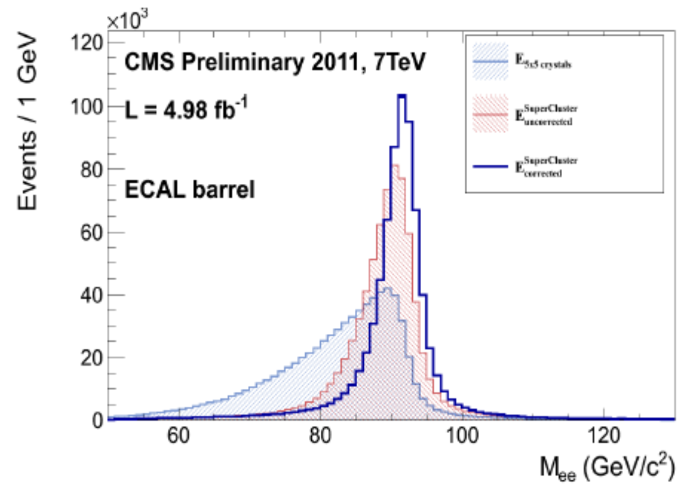
- MVA regression for energy calibration

Photon Energy Corrections, Scale and Resolution

- ECAL cluster energies corrected using a MC trained multivariate regression
  - Improves resolution and restores flat response of energy scale versus pileup
  - Inputs: Raw cluster energies and positions, lateral and longitudinal shower shape variables, local shower positions w.r.t. crystal geometry, pileup estimators
- Regression also used to provide a per photon energy resolution estimate
- **Energy Scale and resolution:** use $Z \to e^+e^-$



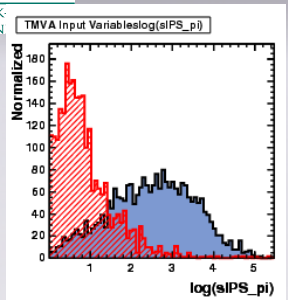
Non converted photons in the barrel |η|<1

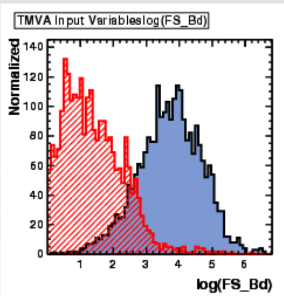


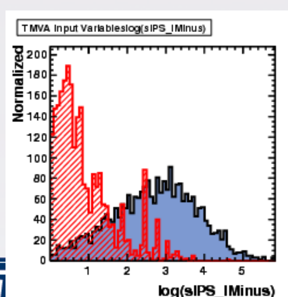
Effect of the regression on the $Z \to e^+e^-$ peak

MVA Classification

- "D" variables (sensitivity to Signal and Background)
- D-dim. variable space → one combined variable

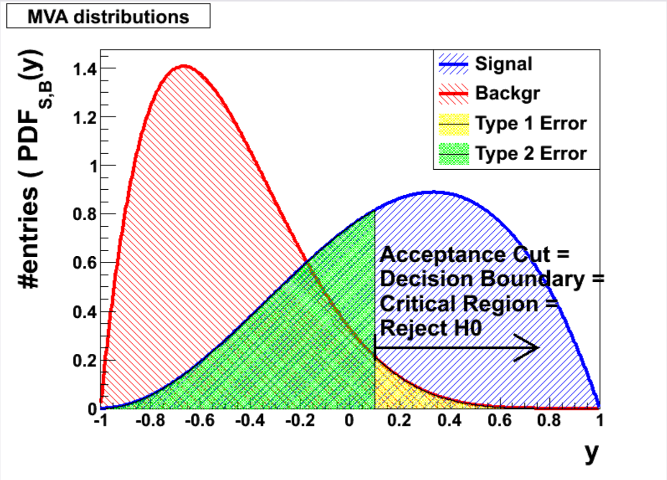






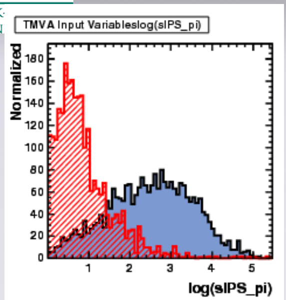
D variable (feature) space

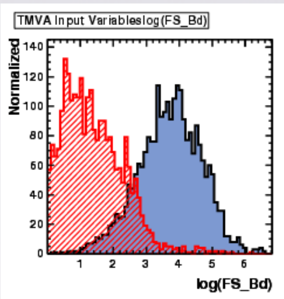
y(x): $R^D \rightarrow R$:

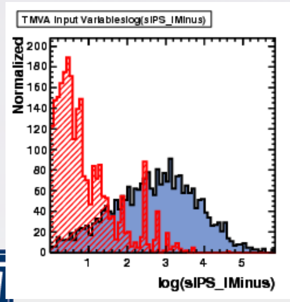


- simple cut on y → complex decision boundary in feature space
- how to find such magical y(x)?

MVA Classification



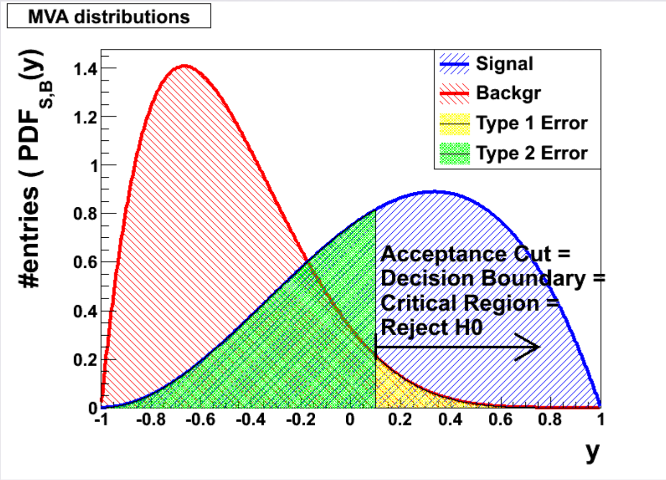




D variable (feature) space

- simple cut on y → complex decision boundary in feature space
- how to find such magical y(x)?

$y(x): R^D \rightarrow R$:

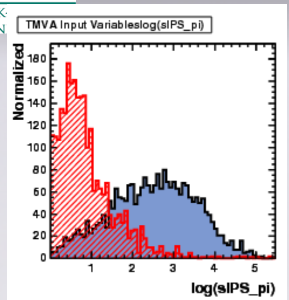


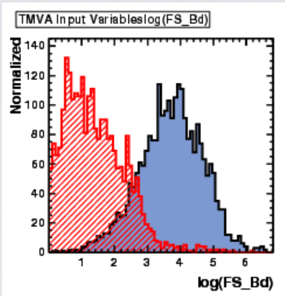
- take model for y(x)
  - linear, nonlinear, piecewiese, flexible, less flexible …
  - fit free parameters to do best (minimize a loss function – i.e. how many misclassified events)

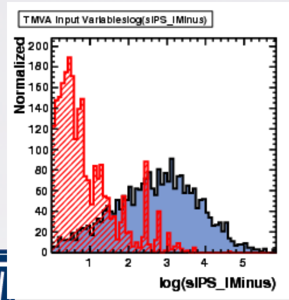
MVA Clas…tion

Regression

Forget about all "blue" and only look at "red" histograms

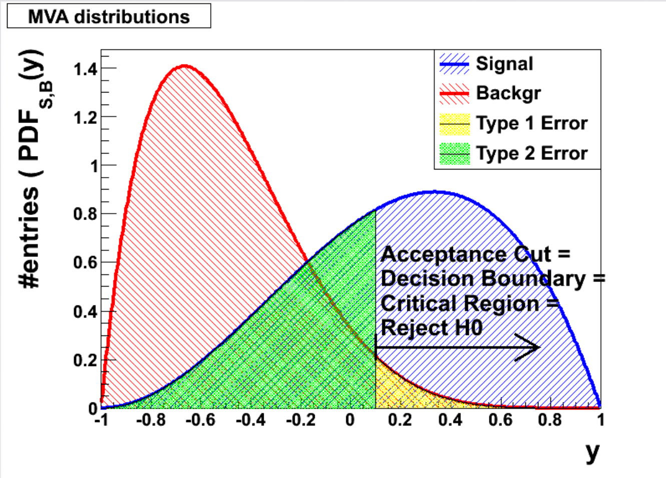






D variable (feature) space

y(x): $R^D \rightarrow R$:



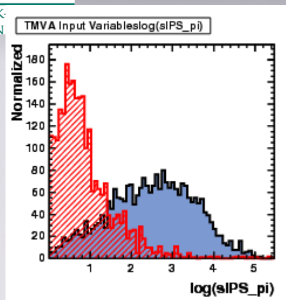
- take model for y(x)
  - linear, nonlinear, piecewiese, flexible, less flexible …
  - fit free parameters to do best (minimize a loss function – i.e. how m… …ents)

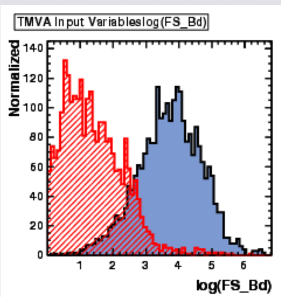
residual $(f(x)$

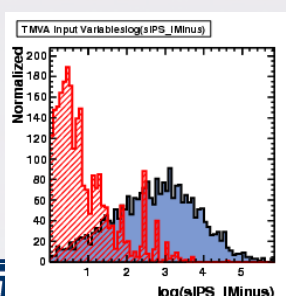
MVA Classification

Regression

Forget about all "blue" and only look at "red" histograms

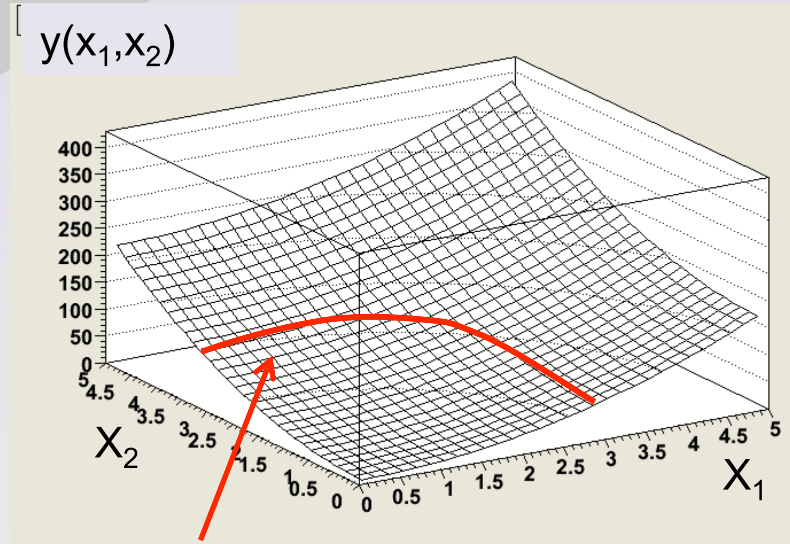






D variable (feature) space

y(x): $R^D \rightarrow R$:



Regression: y=const → contour line

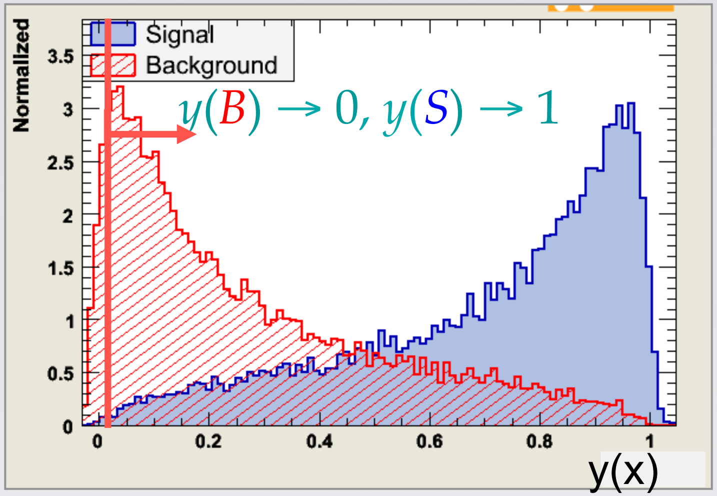
Classification: → decision Boundary

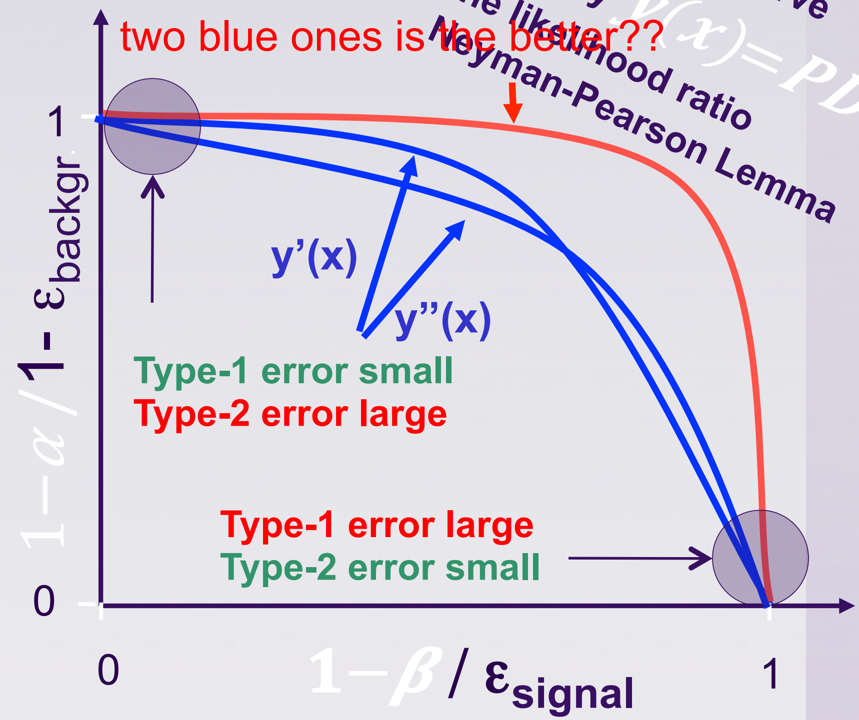
- take model for y(x)
  - linear, nonlinear, piecewiese, flexible, less flexible …
  - fit free parameters to do best (minimize a loss function – i.e. h… …ents)

residual $(f(x)$

Receiver Operation Characteristic (ROC) curve

Signal(H_1) /Background(H_0) discrimination:





Signal(H_1) /Background(H_0) :

- **Type 1 error: reject H_0 although true → background contamination**
  - Significance α: background sel. efficiency **1− α: background rejection**
- **Type 2 error: accept H_0 although false → loss of efficiency**
  - Power: **1− β signal selection efficiency**

What is TMVA doing for you?

- Finds $y(x) : R^n \rightarrow R$
  - given a certain type of model class y(x)
  - "fits" (learns) from events with known type parameters in y(x)
    such that y:
    - CLASSIFICATION: separates well Signal from Background in training data
    - REGRESSION: fits well the target function for training events
  - use for yet unknown events → predictions

→ supervised machine learning

TMVA Content

- **implemented classifiers and regression methods**
  - Rectangular cut optimisation (classification only)
  - Projective and multidimensional likelihood estimator
  - k-Nearest Neighbour algorithm (kNN)
  - LD, Fisher and H-Matrix discriminants (classification only)
  - Function discriminant (classification only)
  - Artificial neural network (MLP) → Bayesian Network features, BFGS
  - Boosted/bagged decision trees (BDT) → RealAdaBoost, Gradient-Boost
  - Rule Fitting (classification only)
  - Support Vector Machine (SVM)
- **implemented data preprocessing stages:**
  - De-correlation, Principal Value Decomposition, Normalisation, "Gaussianisation", Flattening (uniform)
- **combination methods:**
  - Boosting, Categorisation, MVA Committees

Using TMVA

2 main steps:

1. Training phase:

train(build), test and evaluate classifiers using data samples with known signal and background events

2. Application phase:

use to classify unknown data samples

ROOT script for Training

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

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

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

  factory->AddSignalTree         ( (TTree*)input->Get("TreeS") );
  factory->AddBackgroundTree ( (TTree*)input->Get("TreeB") );

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

  factory->PrepareTrainingAndTestTree("", “nTrain_Signal=3000:nTrain_Background=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

← register input variables

← select MVA methods and options

← train, test and evaluate

MAX-PLANCK-INSTITUT FÜR KERNPHYSIK

ROOT script for Application

TMVA

```
void TMVClassificationApplication( )
{
  TMVA::Reader *reader = new TMVA::Reader();

  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 classifier", "weights/MyMVAnalysis_MLP.weights.txt");

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

  // … set branch addresses for user TTree
  for (Long64_t  iev=3000;  iev<theTree->GetEntries(); iev++) {
    theTree->GetEntry(iev);

    var1 = userVar1 + userVar2;
    var2 = userVar1 - userVar2;
    var3 = userVar3;
    var4 = userVar4;

    Double_t   mvaValue =  reader->EvaluateMVA(  "MLP classifier"  );

    // do something with it …
  }
  delete reader;
}
```

- create *Reader*
- register the variables
- read trained classifier(s)
- event loop
- compute input variables
- classifier output

TMVA

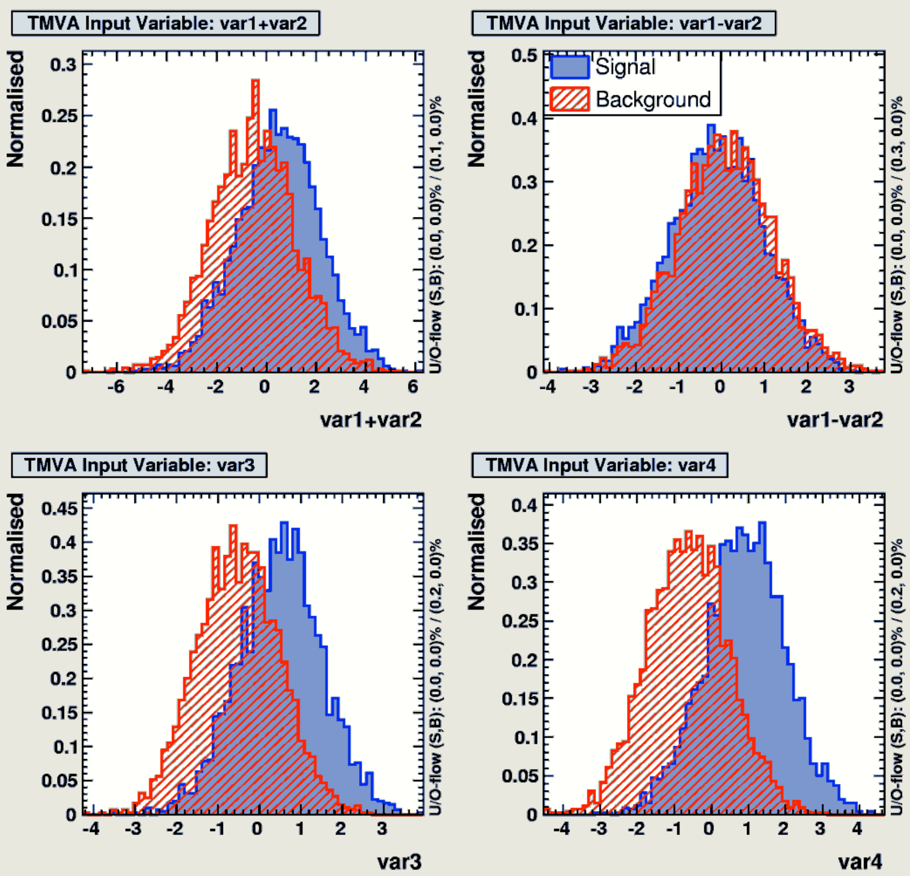
Running TMVA

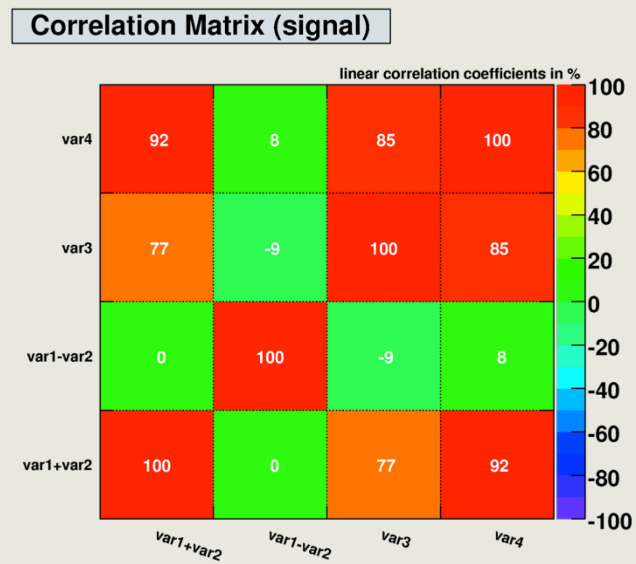
- provide data: ROOT TTree , ASCII-file or event-by-event
- choose variables (or functions ROOT Expressions thereof)
- pre-selection cuts (independent for signal and bkg)
- define global event weights for signal or background input files
- define individual event weight (any variable present in training data)
- choose splitting into training and test samples:
  - Block wise, Randomly, Periodically (i.e. periodically 3 test ev., 2 train ev., etc..)
  - User defined training and test trees
- choose pre-processing of input variables (e.g., de-correlation)
- choose classifiers(s) and it's configuration options
- train/test/evealuate

  → look at the results and diagnostics
- if happy, use trained classifier in the analysis

Toy Example Training:

- Data set with 4 linearly correlated Gaussian distributed variables:



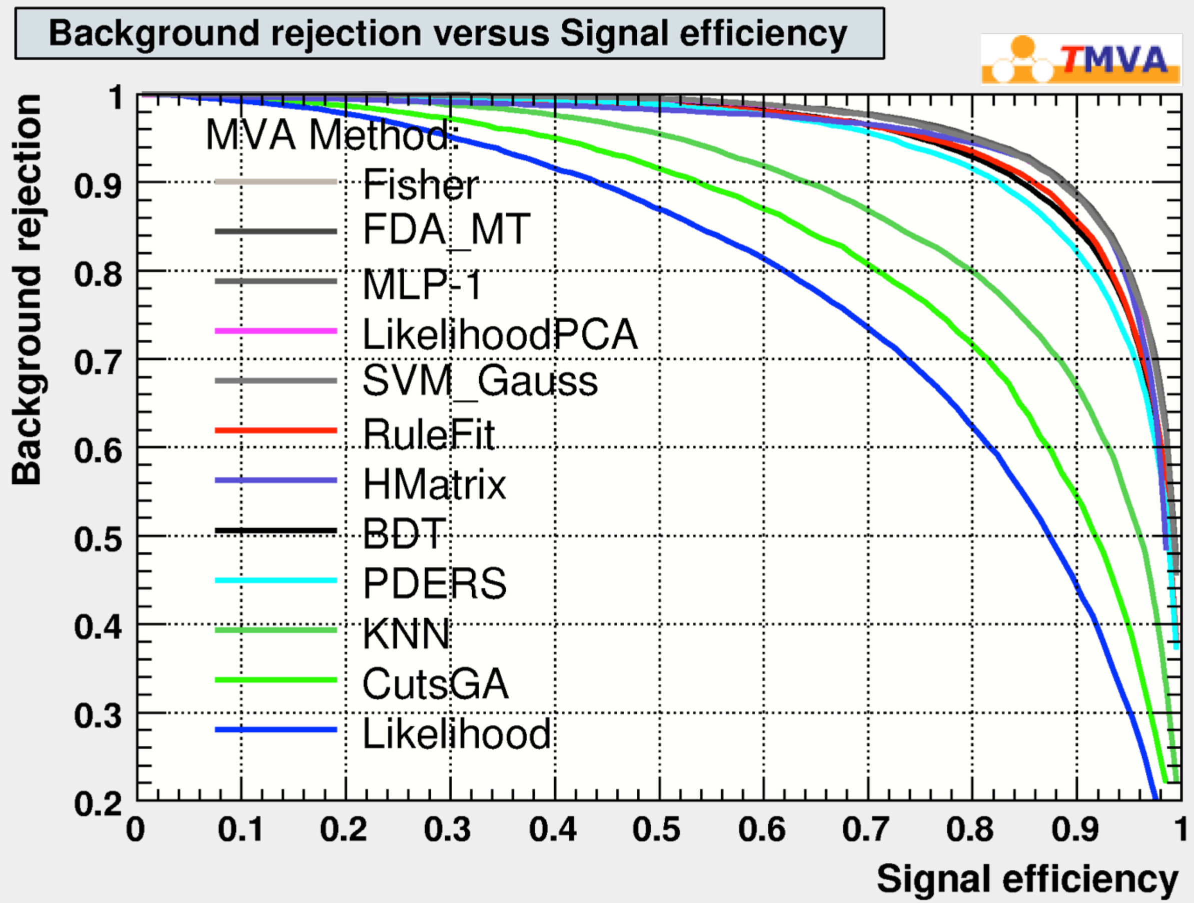



```
--------------------------------------
Rank : Variable   : Separation
--------------------------------------
   1 : var4       : 0.606
   2 : var1+var2  : 0.182
   3 : var3       : 0.173
   4 : var1-var2  : 0.014
--------------------------------------
```

Receiver Operation Characteristics (ROC) Curve

- Smooth background rejection versus signal efficiency curve:
  (from cut on classifier output)

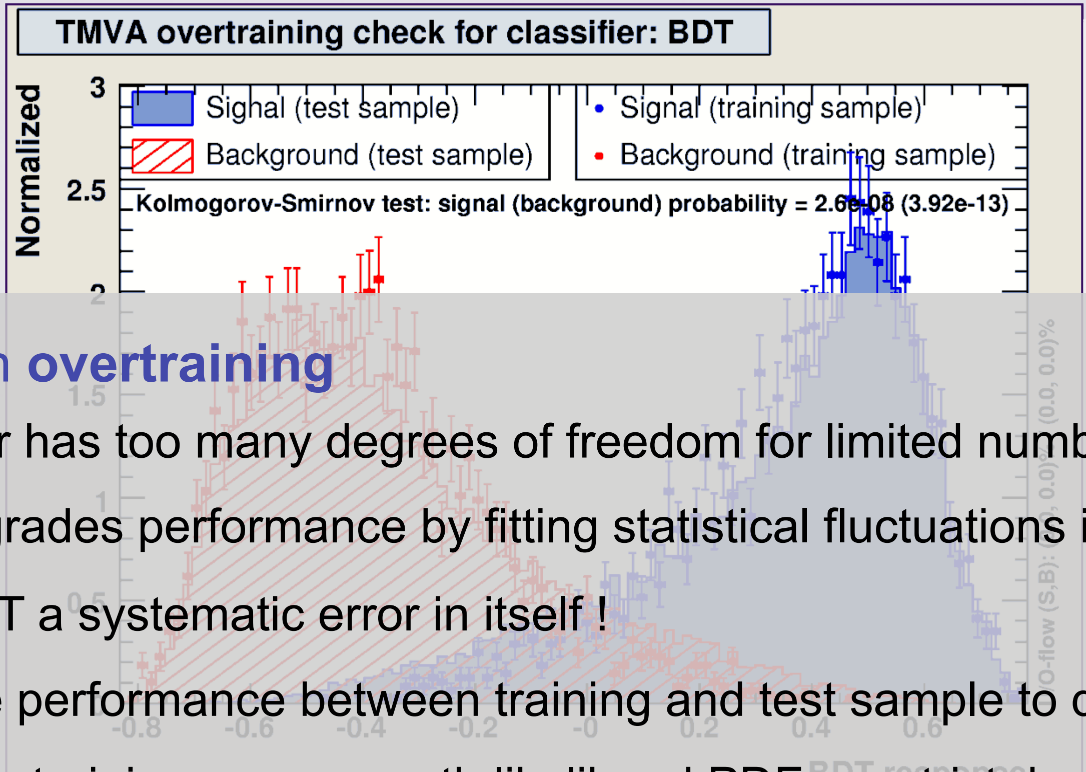
“Specificity” (probability to predict B if true B)



“Sensitivity” (probability to predict S if true S)

Evaluating the Classifier Training

- inspect classifier output distribution
- compare for test *and* training samples …

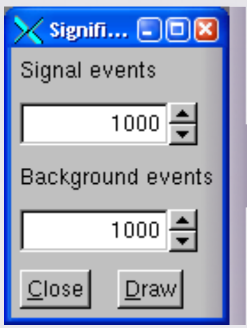


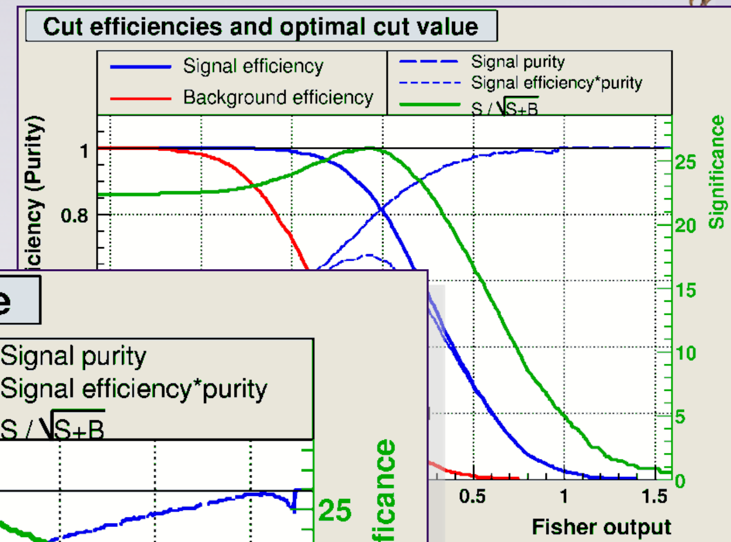
- Remark on **overtraining**
  - classifier has too many degrees of freedom for limited number of training events
    - degrades performance by fitting statistical fluctuations in training sample
    - NOT a systematic error in itself !
  - Compare performance between training and test sample to detect overtraining
  - Avoid overtraining: *e.g.*, smooth likelihood PDFs, restrict decision tree depth, increase kernel parameter size, …

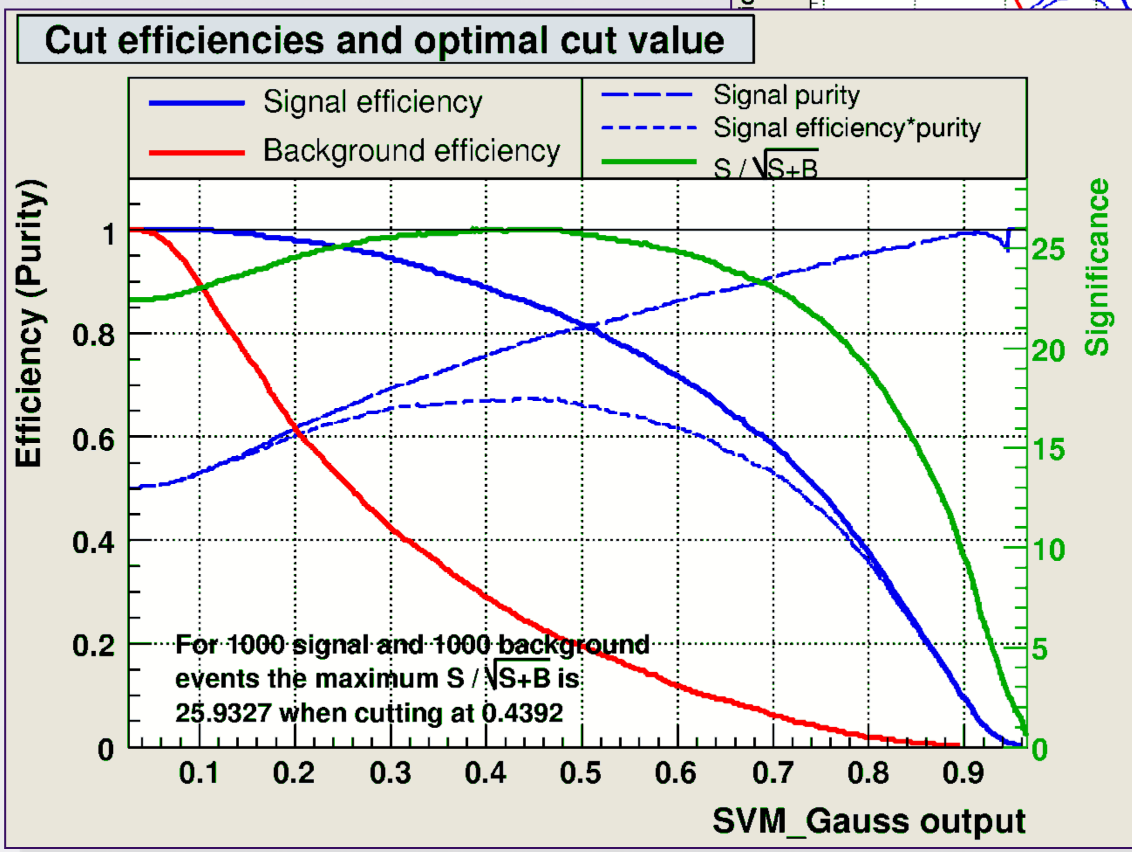
Evaluating the Classifier Training

- Optimal cut for each classifiers …

  Determine the optimal cut (working point) on a classifier output

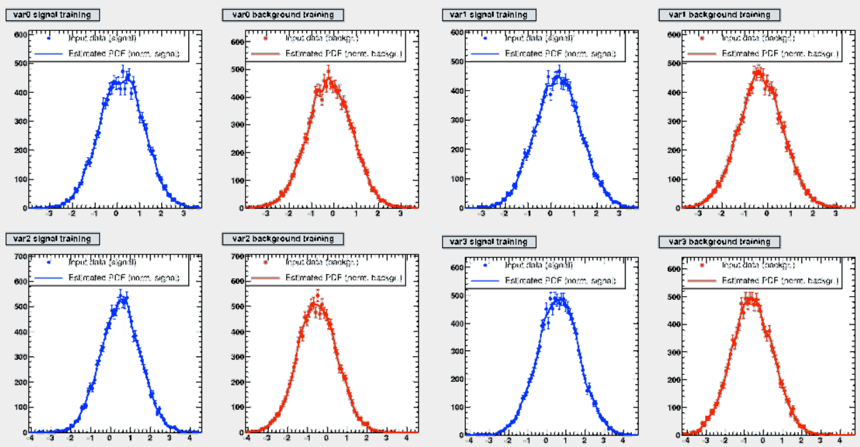


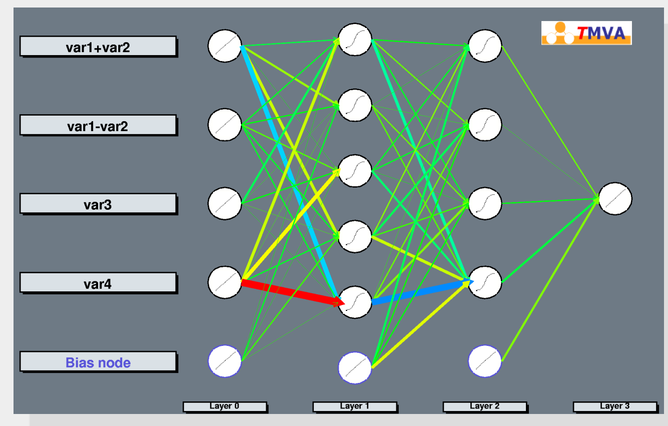


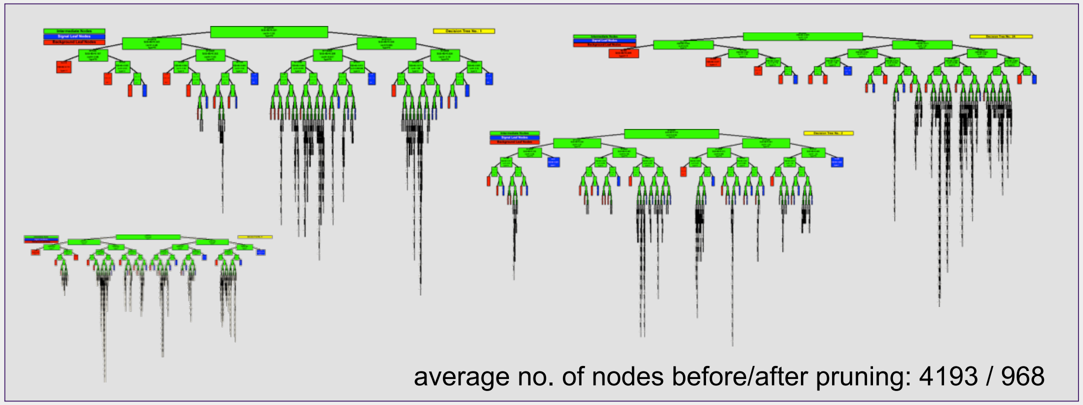


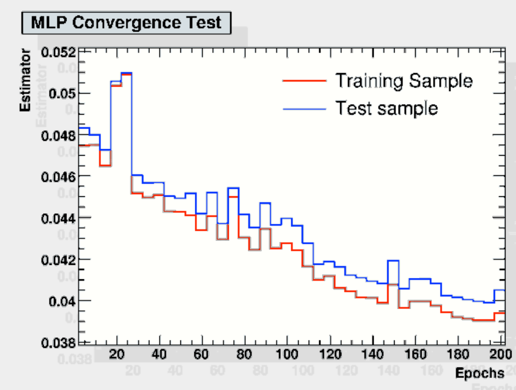
Evaluating the Classifier Training

- Projective likelihood PDFs, MLP training, BDTs, …



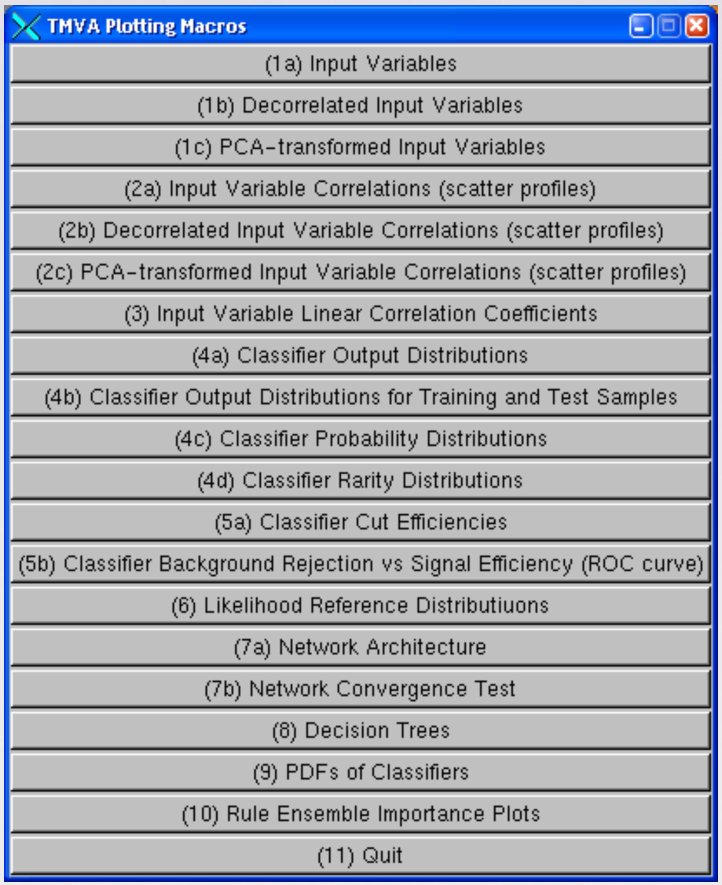






TMVA Evaluation „Framework“

- TMVA helps to understand/optimise your training results !
  - ROOT evaluation/diagnostic scripts (through GUI)



Plot all signal (S) and background (B) input variables with and without pre-processing

Correlation scatters and linear coefficients for S & B

Classifier outputs (S & B) for test and training samples (spot overtraining)

Classifier *Rarity* distribution

Classifier significance with optimal cuts

B rejection versus S efficiency – ROC curve

Classifier-specific plots:
- Likelihood reference distributions
- Classifier PDFs (for probability output and Rarity)
- Network architecture, weights and convergence
- Rule Fitting analysis plots
- Visualise decision trees+ control plots

Evaluating the Classifiers Training (taken from TMVA output…)

Input Variable Ranking

Better variable

```
--- Fisher        : Ranking result (top variable is best ranked)
--- Fisher        : ---------------------------------------------
--- Fisher        : Rank : Variable  : Discr. power
--- Fisher        : ---------------------------------------------
--- Fisher        :    1 : var4      : 2.175e-01
--- Fisher        :    2 : var3      : 1.718e-01
--- Fisher        :    3 : var1      : 9.549e-02
--- Fisher        :    4 : var2      : 2.841e-02
--- Fisher        : ---------------------------------------------
```

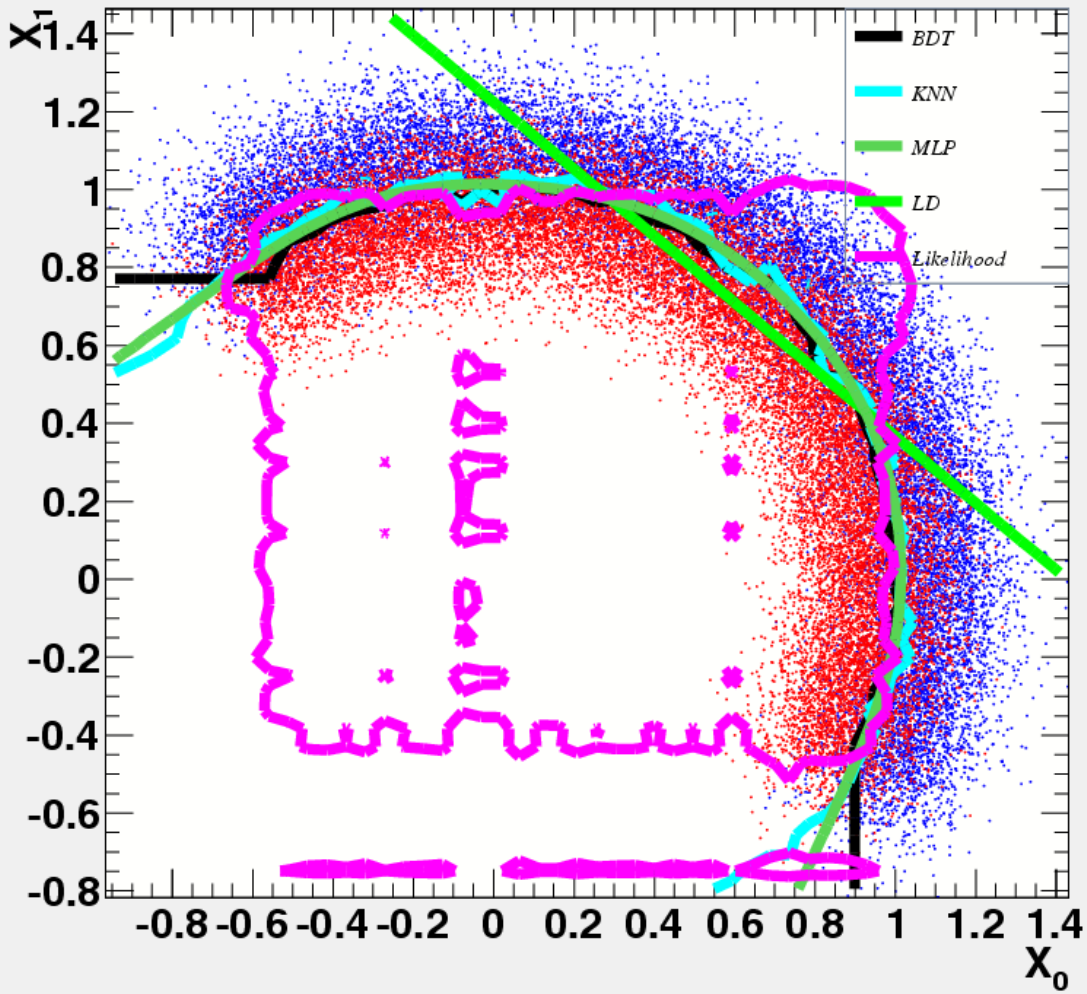
- How discriminating is a variable ? (treat with care, check also the variable separations themselves)

Correlation and Overlap Between Different Classifiers

```
--- Factory       : Inter-MVA overlap matrix (signal):
--- Factory       : ------------------------------
--- Factory       :              Likelihood  Fisher
--- Factory       : Likelihood:      +1.000  +0.667
--- Factory       :     Fisher:      +0.667  +1.000
--- Factory       : ------------------------------
```

- Do classifiers select the same events as signal and background ? If not, there is something to gain !

Decision Boundaries



BDT

kNN

MLP

LD

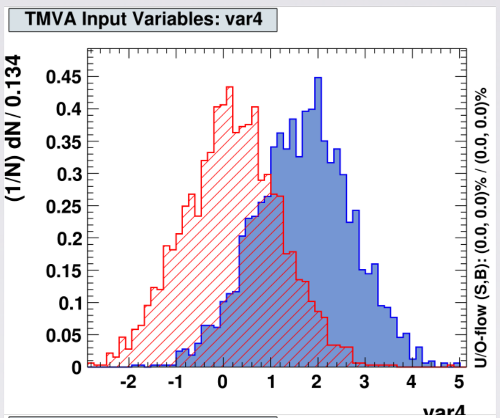
Likelihood

*T*MVA Categories

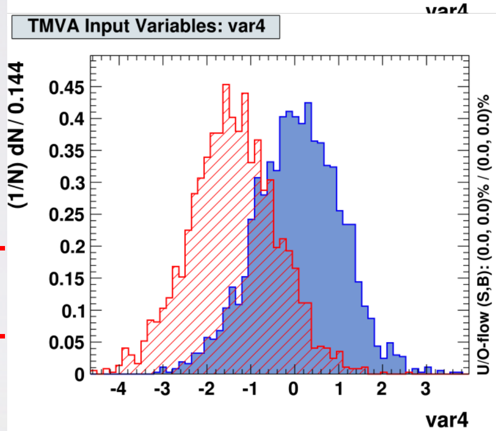
- **TMVA Categories: one classifier per 'region'**
- 'regions' in the detector (data) with different features treated <u>independent</u>
  - improves performance
  - avoids additional correlations where otherwise the variables would be uncorrelated!

Example: var4 depends on some variable "eta"

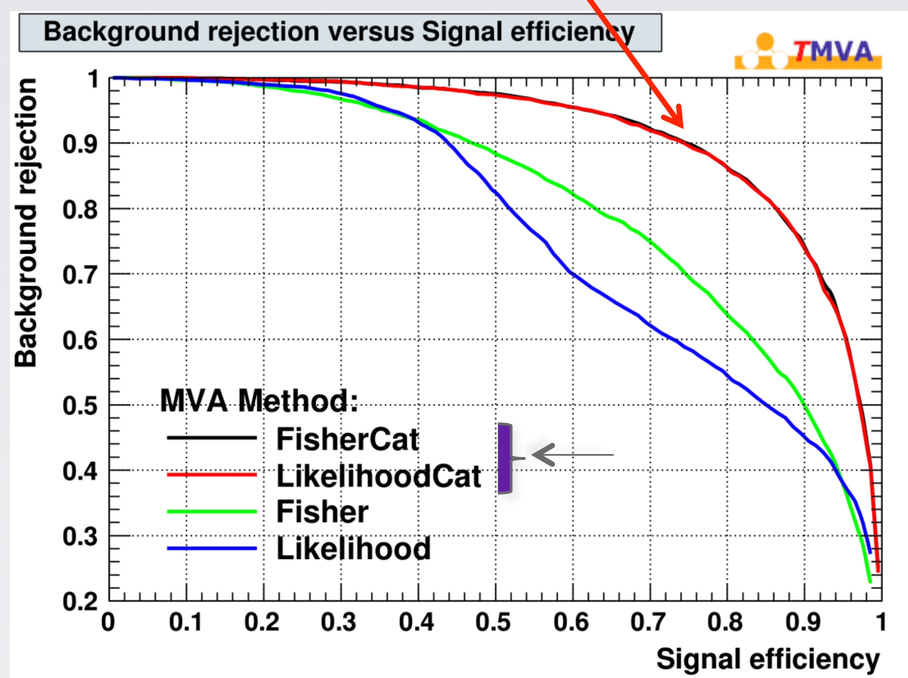
|eta| > 1.3



|eta| < 1.3



Recover optimal performance after splitting into categories



(T)MVA and Systematic Uncertainties

- minimize "systematic" uncertainties (robustness)

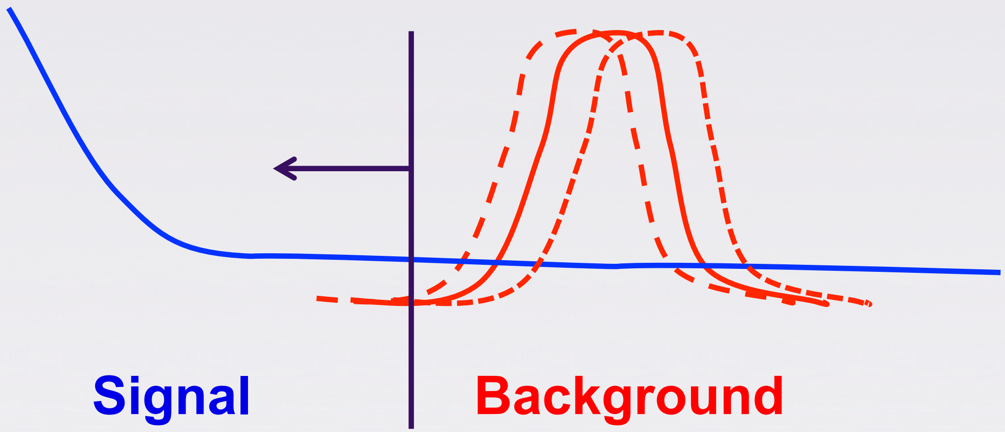
→ "classical cuts" : do not cut near steep edges, or in regions of large sys. uncertainty

→ hard to translate to MVAs:

  → artificially degrade discriminative power (shifting/smearing) of systematically "uncertain" observables IN THE TRAINING

    → remove/smooth the 'edges' → MVA does not try to exploit them

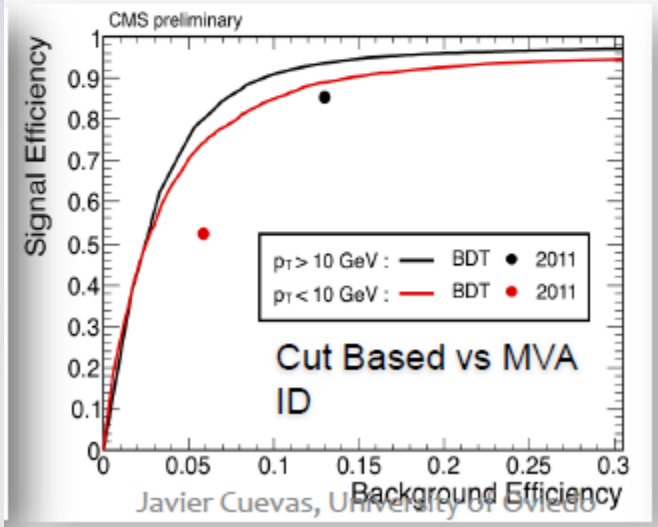
  → First attempts to automatize this are on the way



CMS Higgs Discovery

(such a nice example for MVA usage)

- Multivariate electron identification in 2012
  - ECAL, tracker, ECAL-tracker-HCAL matching and impact parameter (IP) observables



$H \rightarrow \gamma\gamma$

- Analysis selection (MultiVariate Analysis MVA)
  - Vertex ID
    - Input variables: $\Sigma p_T^{2\,(tracks)}$, p_T balance wrt $\gamma\gamma$, conversions information
  - ID photons $p_{T1} > m_{\gamma\gamma} / 3$ $p_{T2} > m_{\gamma\gamma} / 4$
- MVA Diphoton discriminant categories
  - High score
    - signal-like events
    - good $m_{\gamma\gamma}$ resolution
  - Designed to be $m_{\gamma\gamma}$ independent
  - Trained on signal and background MC
  - Input variables:
    - Kinematic variables: $p_{T\gamma} / m_{\gamma\gamma}$, η_γ, $\cos(\varphi_1 - \varphi_2)$
    - Photon ID MVA output for each photon
    - Per-event mass resolutions for the correct and incorrect choice of vertex

PPC 2012 - KIAS - Nov 5th Javier Cuevas, University of Oviedo

LHCb $B_s \rightarrow \mu\mu$ "evidence"

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH (CERN)

LHCb-PAPER-2012-043
CERN-PH-EP-2012-zzz
October 31, 2012

First evidence of the $B_s^0 \rightarrow \mu^+\mu^-$ decay

LHCb collaboration

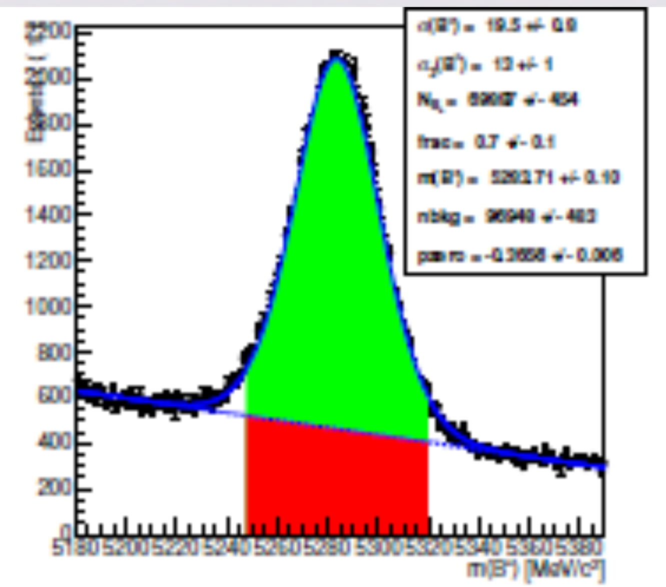
A two-stage multivariate selection, based on boosted decision trees [13], is applied. The first multivariate discriminant, the MVS, removes 80 % of the background while retaining 92 % of signal. The output of the second multivariate discriminant, called BDT in the following, and the dimuon invariant mass are used to classify the selected candidates in a binned two-dimensional space.

You don't even need Monte Carlo to train!

- "sWeights" allow to generate signal and data distributions of any variable that is uncorrelated to one variable where one can fit already a signal +background PDF to.

First observation of the $B^*_{s2}(5840)^0 \to B^{*+}K^-$ decay and properties of the orbitally excited B^0_s mesons

The LHCb collaboration†



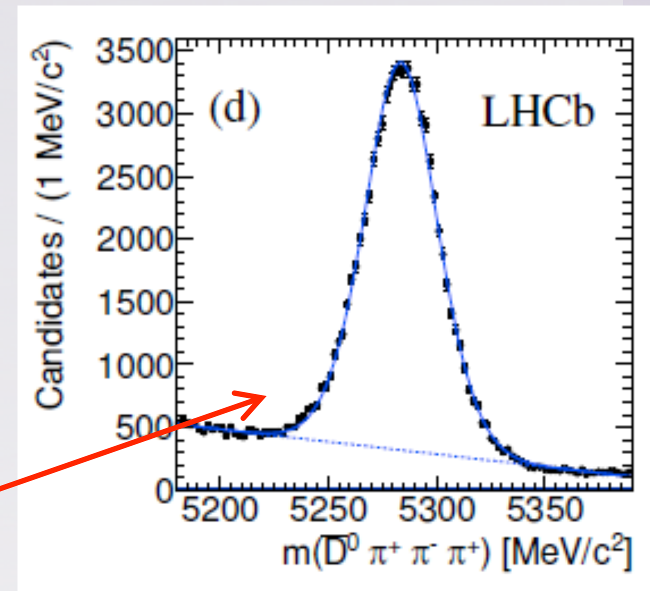
mass distribution

→ fit sig/bkg pdfs

→ sWeights

→ BDT training variables from data

mass distribution after BDT selection



- Negative event weights work FINE in TMVA
  - also BDTs (as proven above) where boosting negative weights is tricky! → we also experiment with alternative "global event pairing"

TMVA Outlook

- **Currently working on:**
  - alternative negative event weight treatment for BDT (and general)
  - user help to increase robustness w.r.t. systematic uncertainties
  - (automatic option parameter tuning)
  - update of Users Guide (ready but valied for ‘next release’ only)
- **Thoughts for the future**
  - constant improvement of the classifiers
    - improve SVM implementation (automatic setting of tuning parameters, training time estimate, ..) and promote its usage
    - speed up ?
  - start thinking about “parallelisation” : starting points
    - serialize processing of different methods → should be easy to parallelise for different CPU cores
    - BDT training (each node split →loops serially over each variable)
    - Genetic fitting algorithm
    - GPU usage ? → one user already tried, need to follow up
    - PROOF-lite ? (well, I’m one of those that …)

Summary

- **Multivariate Classifiers (Regressors)**

  → (fit) decision boundary (target function)
- **TMVA provides:**
  - May different machine learing algorithms, all easily accessible witht the same e.g.
    - PDF based: multi-dimensional (and projective) Likelihood
    - Linear: Linear Classifier (e.g. Fisher Discriminant)
    - Non-Linear: ANN, BDT, SVM
- **TMVA helps to understand/judge/improve your training**
  - carefully study the control plots
  - compare different MVAs !
  - find working point on ROC curve
- **MVAs are not magic:**
  - systematic uncertainties don't lie in the training !!
  - estimate them similar as you'd do in classical cuts

Generalities for (T)MVA Analyses

- There is no magic in MVA-Methods:
  - “black boxes” ? → they are not sooo hard to understand
  - you typically still need to make careful tuning and do some “hard work”
  - no “artificial intelligence” … just “fitting decision boundaries” in a given model

- The most important thing at the start is finding good observables
  - good separation power between S and B
  - little correlations amongst each other
  - watch correlation between selection variables and the parameters you try to measure!

- Think also about possible combination of variables
  - this may allow you to eliminate correlations
    - rem.: you are MUCH more intelligent than what the algorithm will do

Generalities for (*T*)MVA Analyses

- Apply pure preselection cuts and let the MVA only do the difficult part.

- "Sharp features should be avoided" → numerical problems, loss of information when binning is applied
  - simple variable transformations (i.e. log(variable)) can often smooth out these areas and allow signal and background differences to appear in a clearer way

Systematic Uncertainties

- Multivariate Classifiers THEMSELVES don't have systematic uncertainties
  → even if trained on a "phantasy Monte Carlo sample"
  - there are only "bad" and "good" performing classifiers !
    - OVERTRAINING is NOT a systematic uncertainty !!
    - difference between two classifiers resulting from two different training runs DO NOT CAUSE SYSTEMATIC ERRORS
  - same as with "well" and "badly" tuned classical cuts
  - MVA classifiers: → only select a region(s) in observable space
- Efficiency estimate (Monte Carlo) → statistical/systematic uncertainty
  - involves "estimating" (uncertainties in) distribution of $PDF_{y_{S(B)}}$

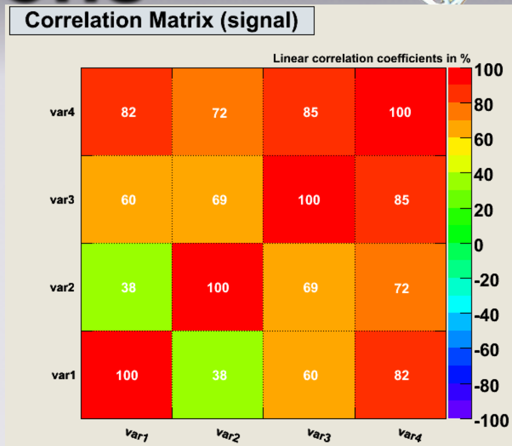
  → estimate systematic error/uncertainty on efficiencies
    - statistical "fluctuations" → re-sampling (Bootstrap)
    - "smear/shift/change" input distributions and determine $PDF_{y_{S(B)}}$
    - simple "cut variation" has never been the best test ☺
- Only involves "test" samples…
  - systematic uncertainties have nothing to do with the training !!

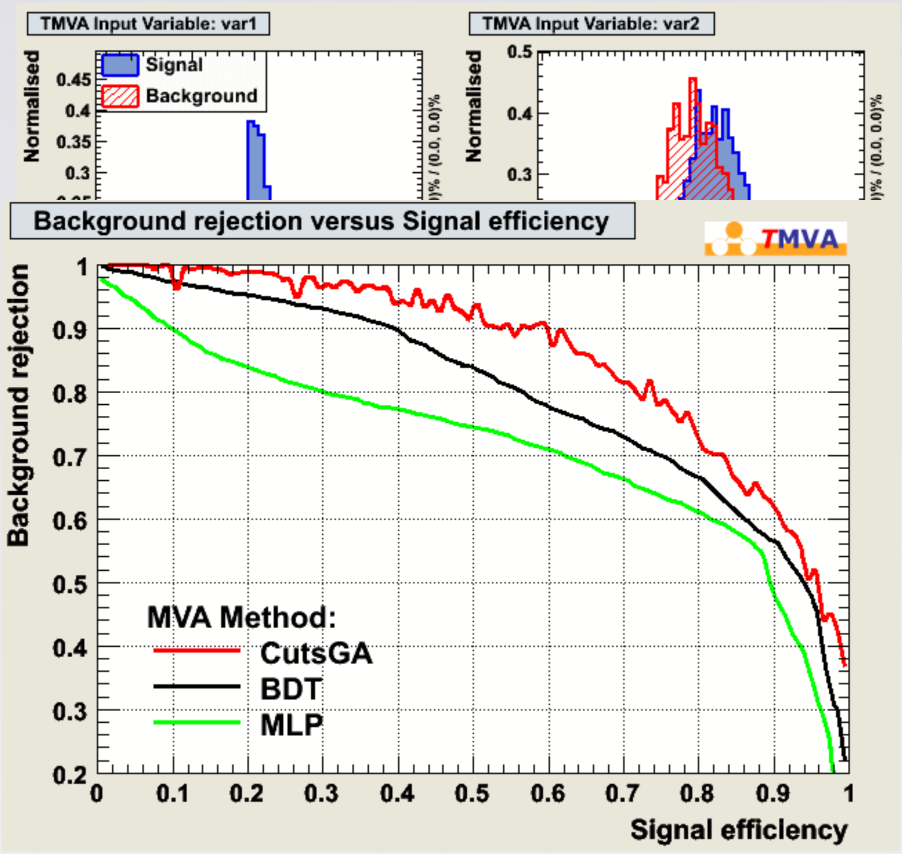
(T)MVA and Systematic Uncertainties

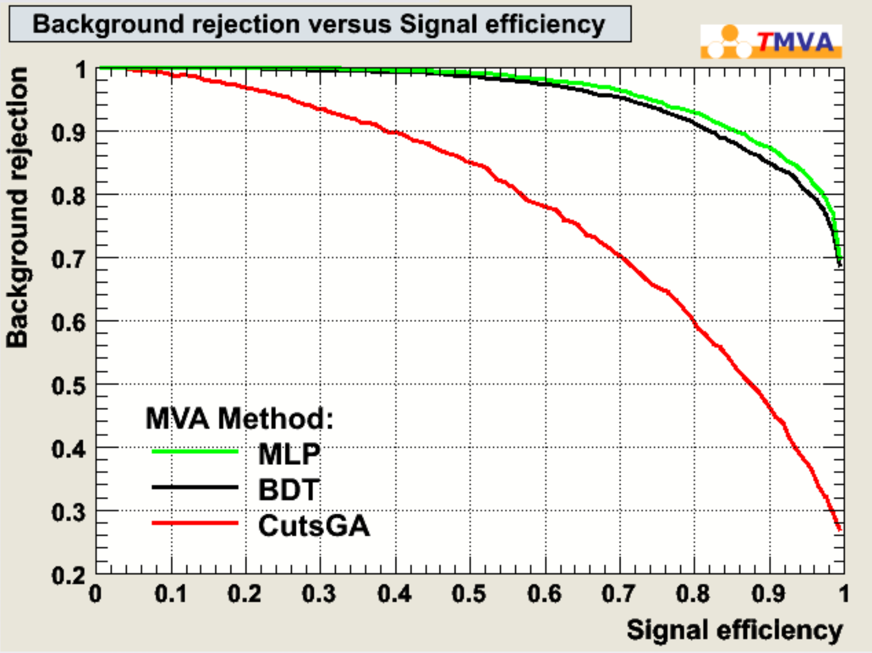
- → Don't be afraid of correlations!
  - → typically "kinematically generated" → easily modeled correctly
  - → "classical cuts" are also affected by "wrongly modeled correlations"
  - → MVA method let's you spot mis-modeled correlations!
    - → "projections" of input variables
    - → + the combined MVA test statistic "y(x)" !

Systematic "Error" in Correlations

- Use as training sample events that have correlatetions
  - optimize CUTs
  - train an propper MVA (e.g. Likelihood, BDT)



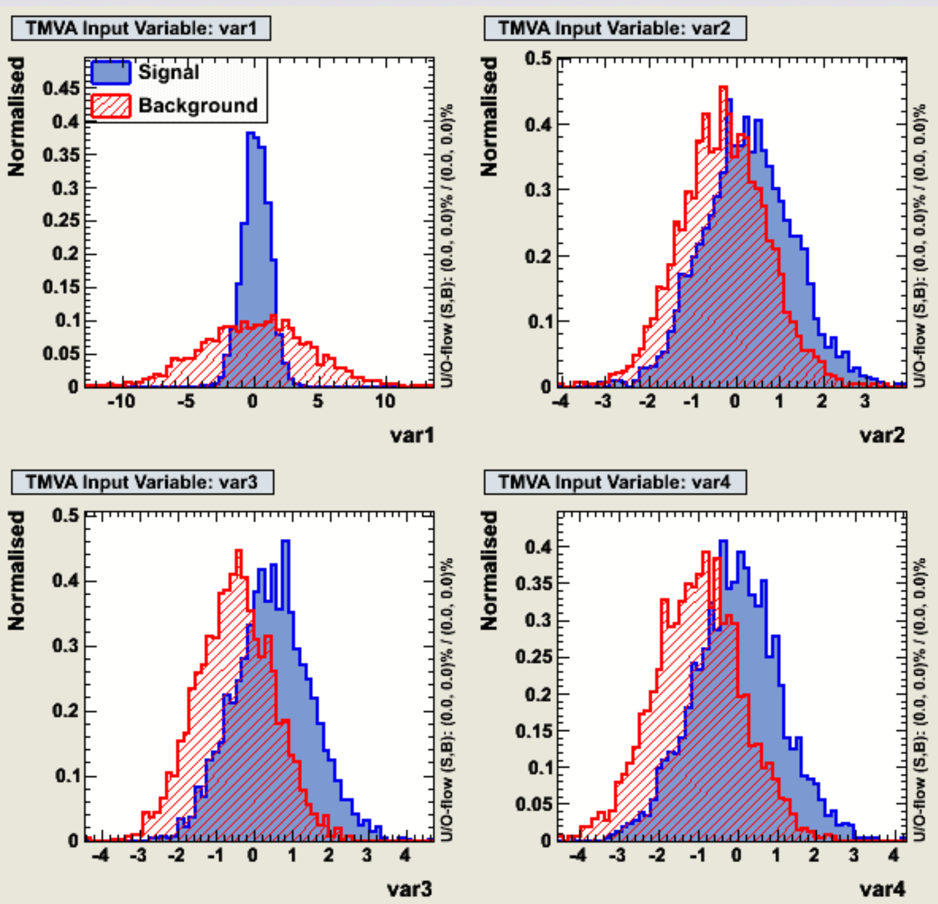


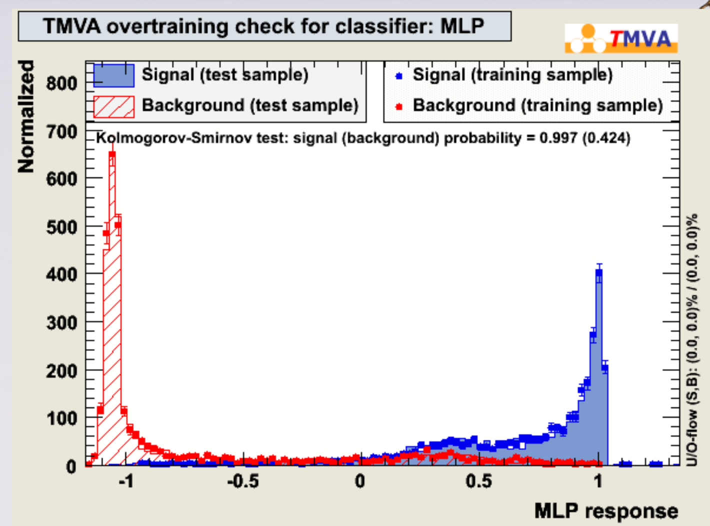


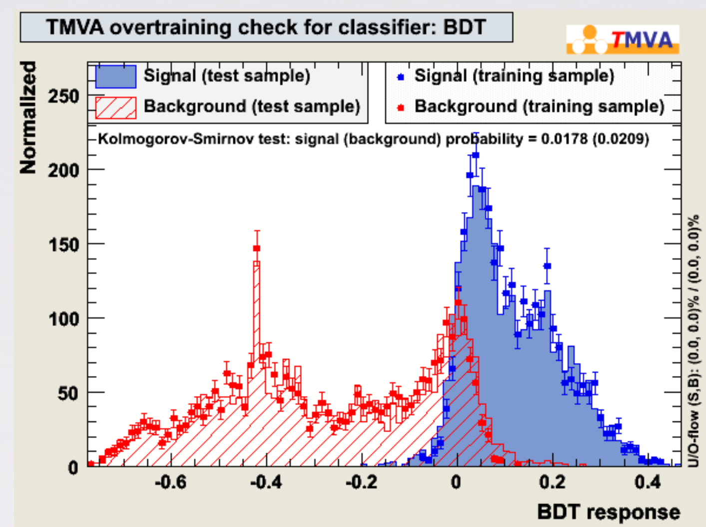
- Assume in "real data" there are NO correlations → SEE what happens!!

Systematic “Error” in Correlations

- Compare “Data” (TestSample) and Monte-Carlo
- both taken from the same underlying distributions

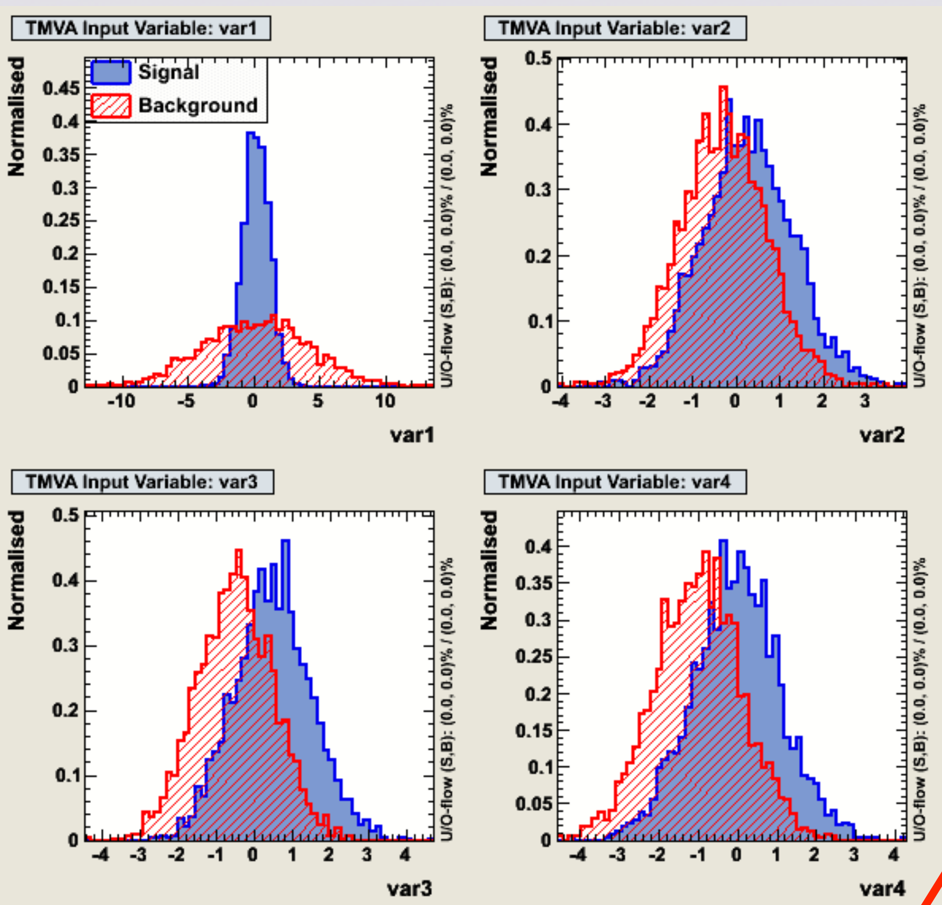


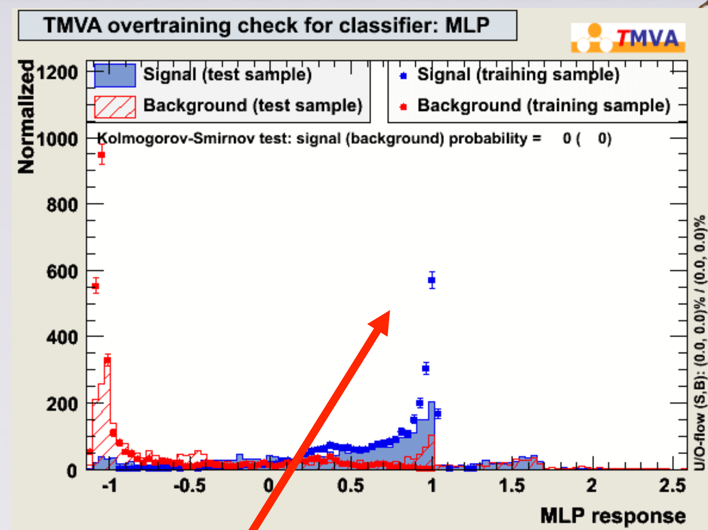


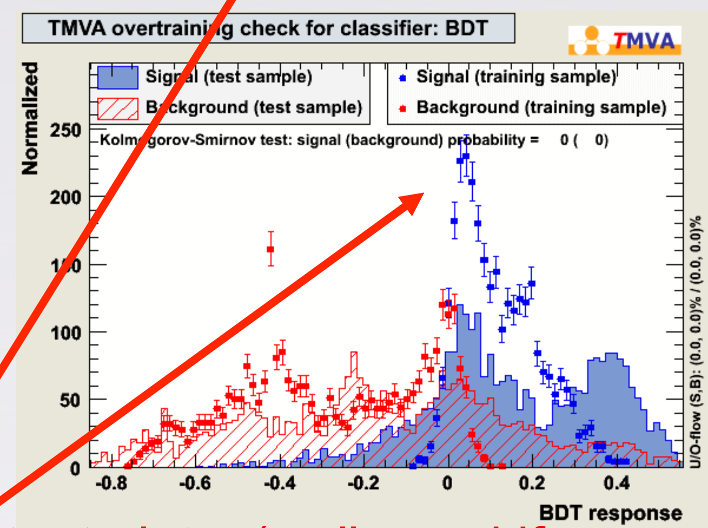


Systematic "Error" in Correlations

- Compare "Data" (TestSample) and Monte-Carlo
- both taken from the same underlying distributions that differ by the correlation!!!







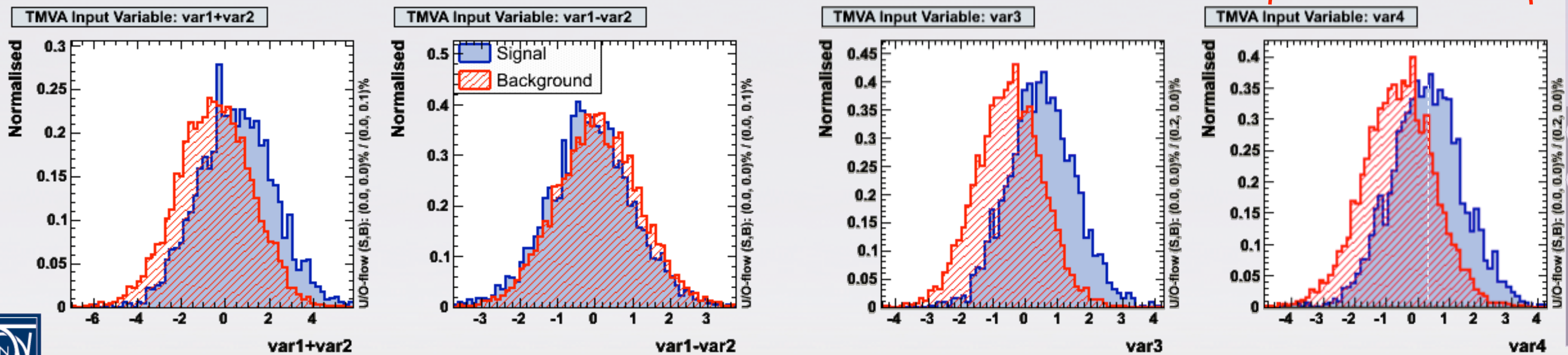
Differences are ONLY visible in the MVA-output plots (well…and if you were to study the 'cut sequences')

Robustness Against Systematics

- Is there a strategy however to become 'less sensitive' to possible systematic uncertainties
  - i.e. classically: variable that is prone to uncertainties → do not cut in the region of steepest gradient
  - classically one would not choose the most important cut on an uncertain variable
- Try to make classifier less sensitive to "uncertain variables"
  - i.e. re-weight events in training to decrease separation
  - in variables with large systematic uncertainty

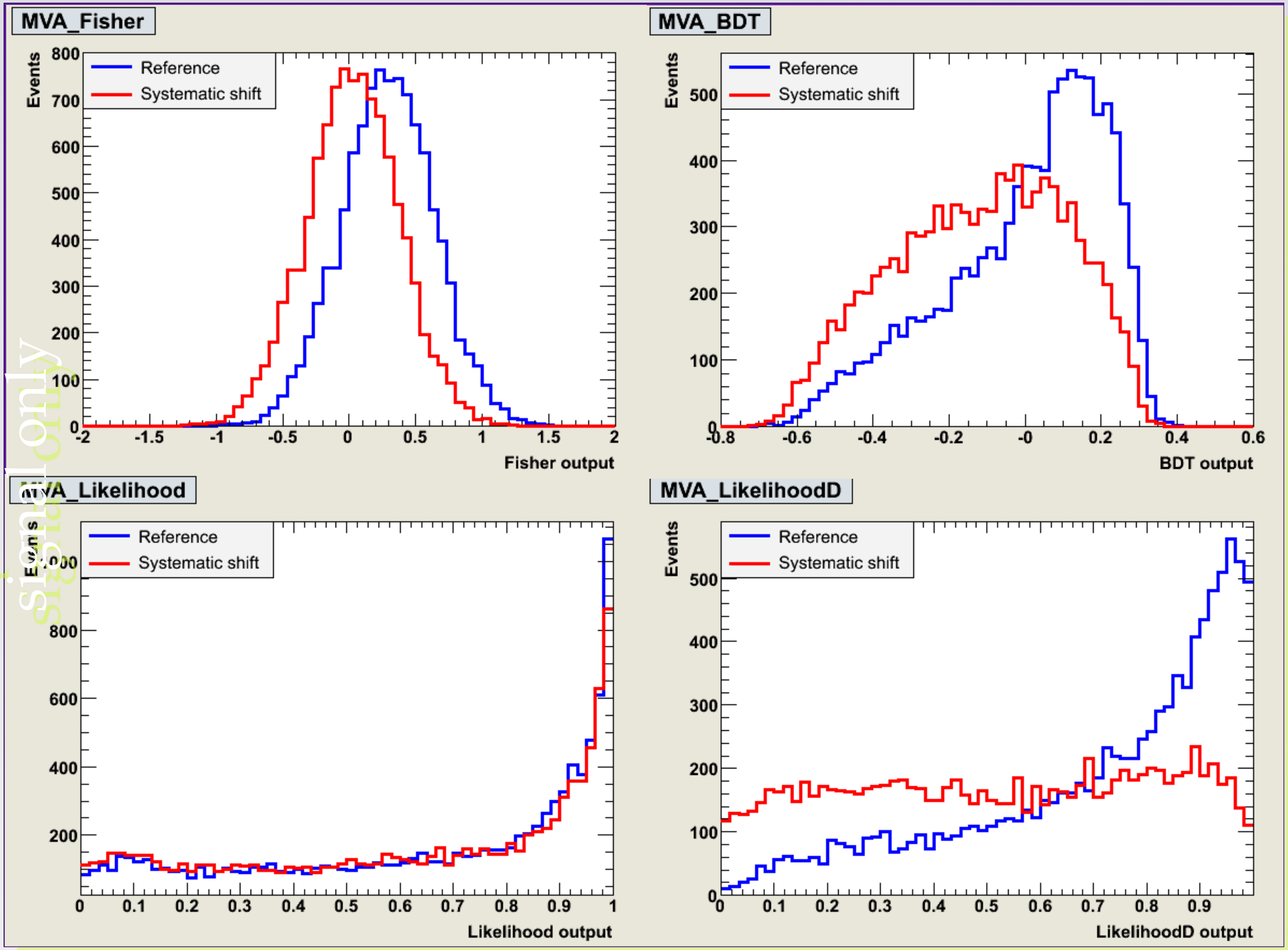
(certainly not yet a recipe that can strictly be followed, more an idea of what could perhaps be done)

"Calibration uncertainty"
→ possible shift
→ worsen (or increase) the discrimination power of "var4"



Robustness Against Systematics

Classifier output distributions for signal only



Robustness Against Systematics

Classifier output distributions for signal only