TMVA – Toolkit for Multivariate Analysis

Jörg Stelzer(*) – DESY, Hamburg

ACAT 2008, 3rd-7th Nov, Erice, Sicily

(*)For the TMVA developer team: A. Höcker, P.Speckmayer, J.Stelzer, H.Voss
and many contributors

MVA in HEP

In a time of ever larger dataset with an ever *smaller signal* fraction it becomes increasingly important to use *all the features* of signal and background data that are present. That means that not only the variable distributions of the signal must be evaluated against the background, but in particular the information hidden in the *variable correlations* must be explored. The possible complexities of the data distributed in a *high-dimensional space* are difficult to disentangle manually and without the help of automata.

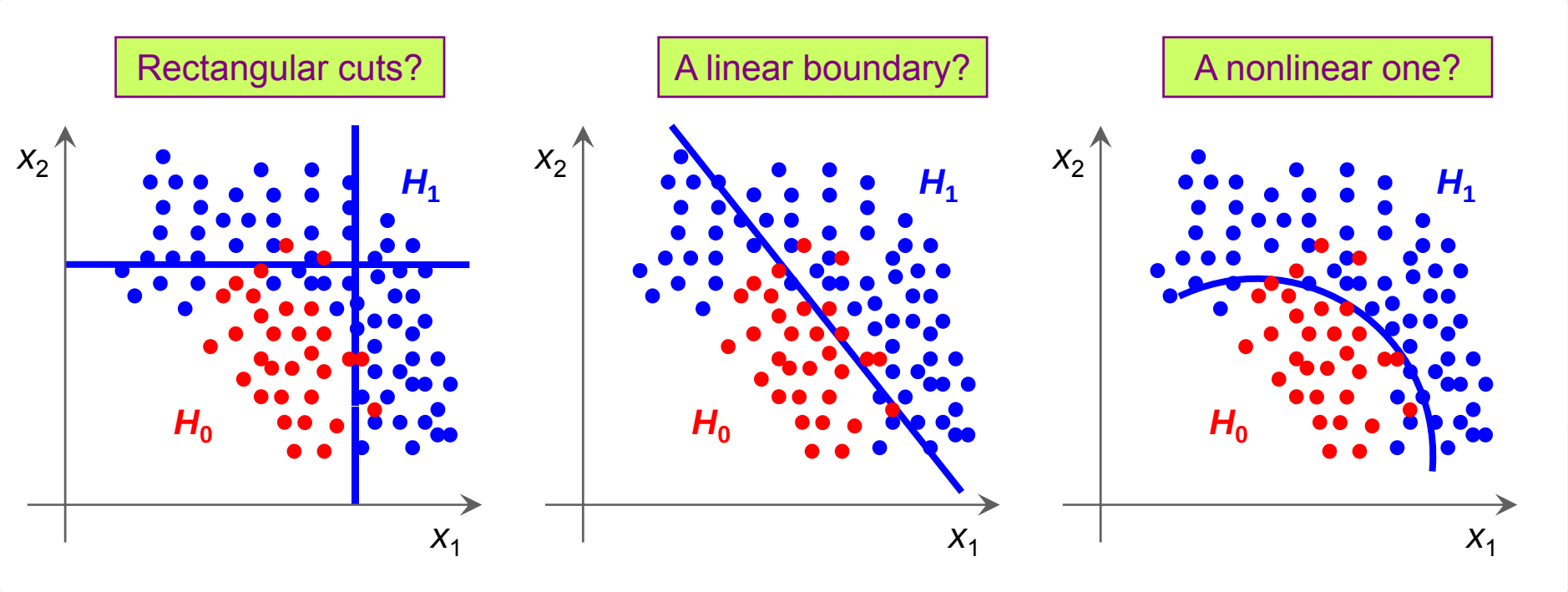
➡ Machinated multivariate analysis needed for data analysis in High Energy Physics

Outline of the Presentation

- The basics of multivariate analysis?
- A short introduction to the idea of TMVA and its history
- Quick survey of available classifiers
  - And how to judge their performance
- An outlook to TMVA 4

Event Classification

- Suppose data sample with two types of events: H_0, H_1
  - We have found discriminating input variables x_1, x_2, …
  - What decision boundary should be used to separate events of type H_0 and H_1 ?

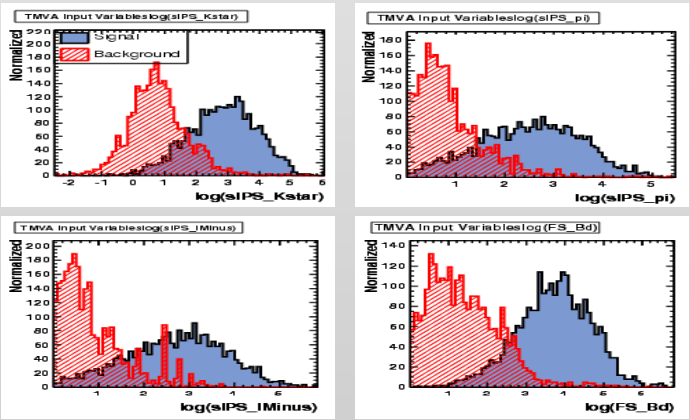


- How can we decide this in an *optimal* way ? → **Let the machine do it using Multivariate Classifiers!**

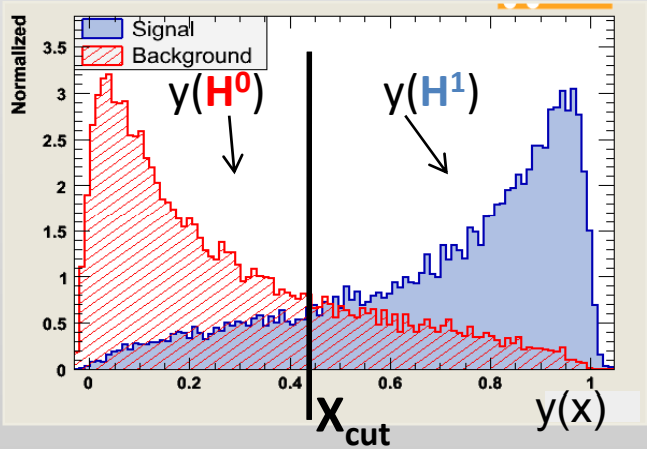
Multivariate (Binary) Classifiers

- All multivariate binary classifiers have in common to condense (correlated) multi-variable **input information** in a **single scalar output variable**
  - Regression problem from N-dimensional feature space into 1 dimensional real number **y: $R^N \rightarrow R$**

R^N



R



- <u>Classification</u>: Mapping from R to the class (binary: signal or background)
  - Definition of a "*cut*": *signal* $y>=X_{cut}$, *background* $y<X_{cut}$
  - Choice of cut driven by the type of analysis

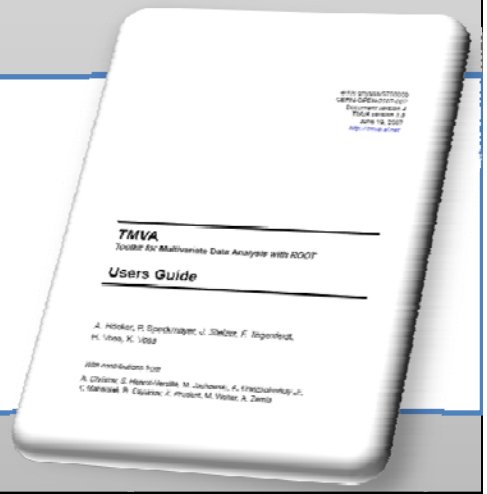
- Cut classifier is an exception: Direct mapping from $R^N \rightarrow$ {Signal,Background}

TMVA – A Simple Idea

- Large number of classifiers exist in different places and languages
  - Neural net libraries, BDT implementations, Likelihood fitting package

- ***T*MVA**-approach: rather than just re-implementing MVA techniques in ROOT:
  - Have one common platform / interface for all MVA classifiers
  - Have common data pre-processing capabilities
  - Provide common data input and analysis framework (ROOT scripts)
  - Train and test all classifiers on same data sample and evaluate consistently
  - Classifier application w/ and w/o ROOT, through macros, C++ executables or python

- TMVA is hosted on SourceForge http://tmva.sf.net/ (mailing list)
- Integrated in ROOT since ROOT v5.11/03
- Currently 4 core developers, and many active contributors
- Users guide ***arXiv physics/0703039***



Features of the TMVA Data Handling

- Data input format: ROOT TTree or ASCII
- Supports selection of any subset or combination or function of available variables and arrays (var1="sin(x)+3*y", var2="…") just like ROOT's TTree::Draw()
- Supports application of pre-selection cuts (possibly independent for signal and bkg)
- Supports global event weights for signal or background input files
- Supports use of any input variable as individual event weight
- Supports various methods for splitting data into training and test samples:
  - Block wise, randomly, periodically
  - User defined training and test trees
- Preprocessing of input variables (*e.g.*, decorrelation, PCA)
  - Improves performance of projective Likelihood

Quick Overview Of The Methods

Conventional Linear Classifiers

- Cut based (still widely used since transparent)
- Projective likelihood estimator (optimal if no correlations)
- Linear Fisher Discriminant (robust and fast)

Still Popular!
Should only be used if non-linear correlations are absent

Common Non-linear Classifiers

- Neural Network (very powerful, but training often ends in local minima)
- PDE: Range Search, kNN, Foam (multi-dim LHood → optimal classification)
- Function Discriminant Analysis

Modern classifiers recent in HEP

- Boosted Decision Tree (brute force, not much tuning necessary)
- Support Vector Machine (one local minima, careful tuning necessary)
- Learning via rule ensembles

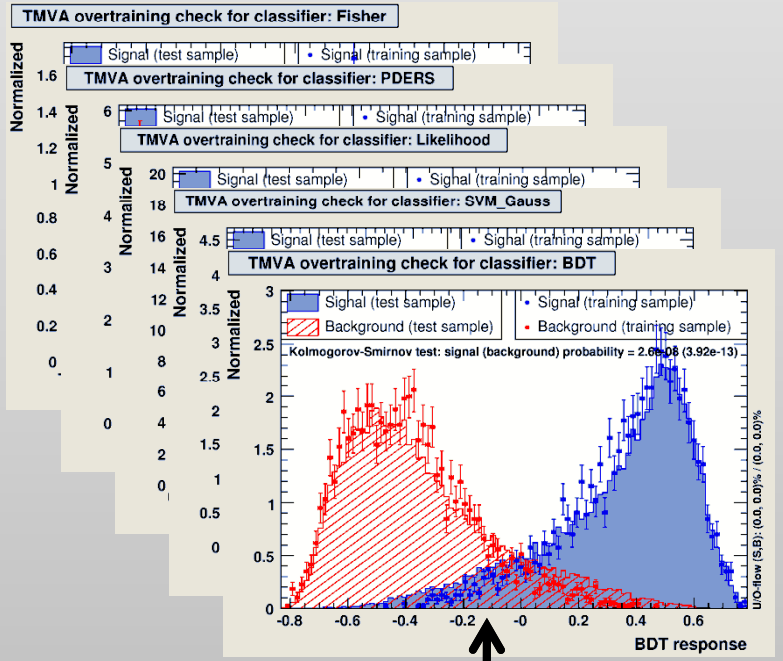
Detailed description of all classifiers and references in the Users Guide!

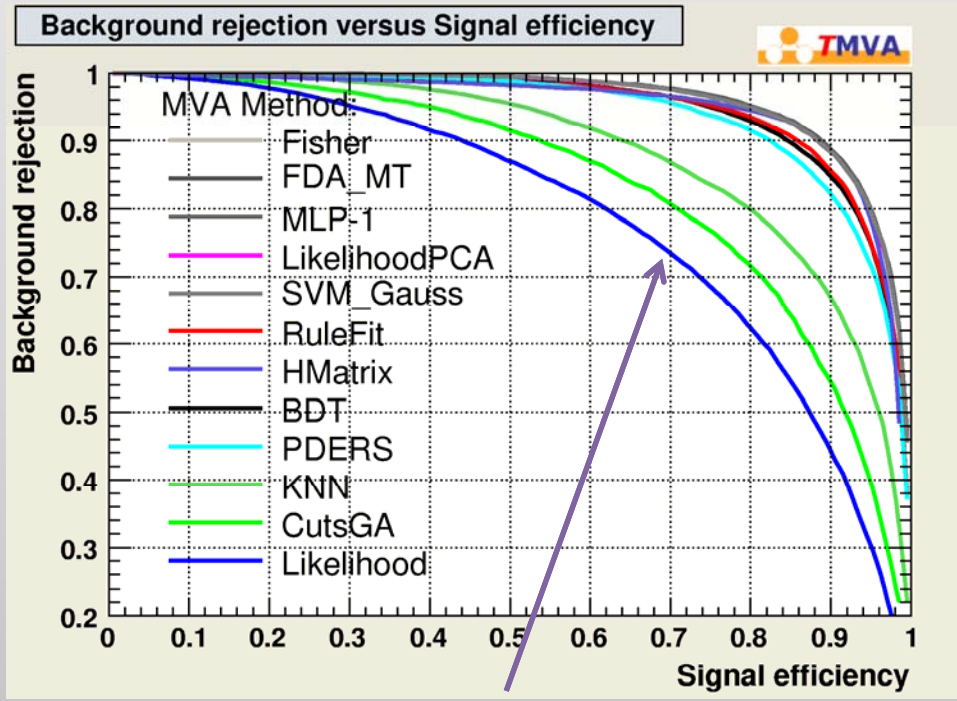
Evaluation

- Training of classifiers: Users Guide, presentations (http://tmva.sf.net/talks.shtml) or tutorial (https://twiki.cern.ch/twiki/bin/view/TMVA/WebHome)
- New web-page: *classifier tuning parameters at* http://tmva.sf.net/optionRef.html

Evaluation:

Classifier output variable (test sample)



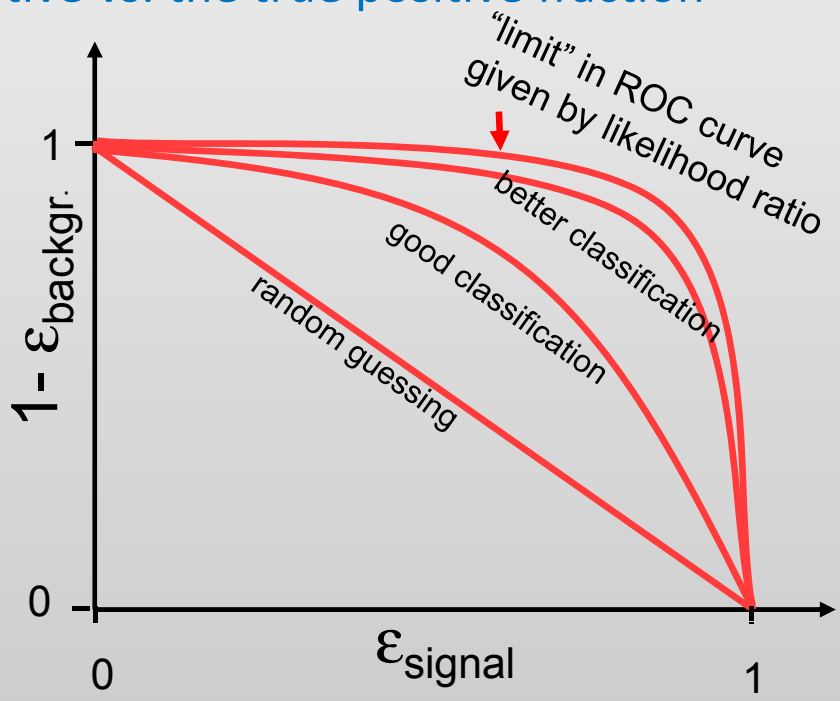


ROC Curve

Scan over classifier output variable creates set of (ε_{sig},ε_{bkgd}) points

The ROC Curve

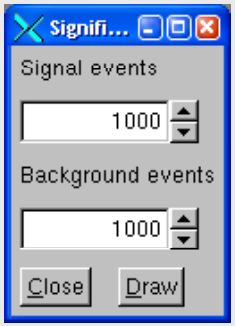
- ROC (receiver operating characteristics) curve describes performance of a binary classifier by plotting the false positive vs. the true positive fraction

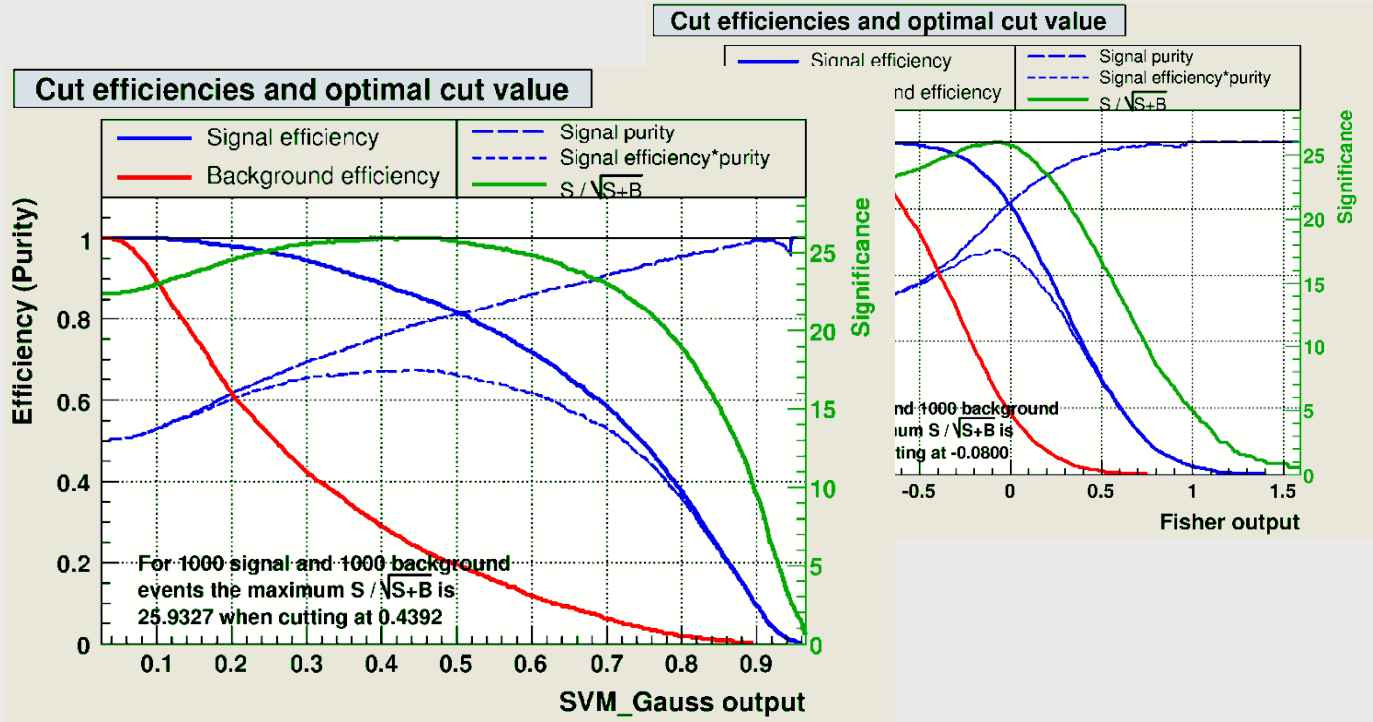


- False positive (type 1 error) = ε_{backgr}:
  - classify background event as signal
  - → loss of purity
- False negative (type 2 error) = $1-\varepsilon_{signal}$:
  - fail to identify a signal event as such
  - → loss of efficiency
  - True positive: ε_{signal}
- Likelihood ratio test $y(x) = \dfrac{P(S \mid x)}{P(B \mid x)}$
- The Likelihood ratio used as "selection criterion" y(x) gives for each selection efficiency the best possible background rejection (Neyman-Pearson)
  - It maximizes the area under the ROC-curve (PDE classifiers)

Optimal Cut for Each Classifier

- Working Point: Optimal cut on a classifier output (=optimal point on ROC curve) depends on the problem
  - Cross section measurement: maximum of S/√(S+B)
  - Search: maximum of S/√(B)
  - Precision measurement: high purity
  - Trigger selection: high efficiency





No Single Best

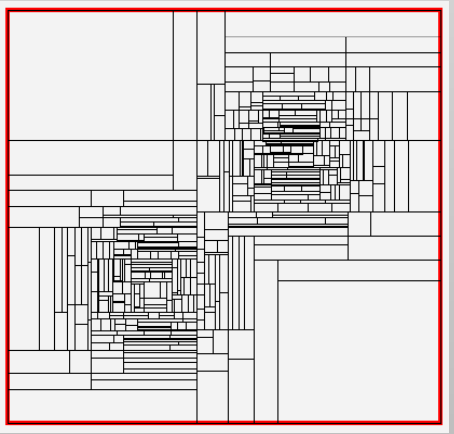
Criteria		Classifiers								
		Cuts	Likeli-hood	PDERS/ k-NN	H-Matrix	Fisher	MLP	BDT	RuleFit	SVM
Perfor-mance	no / linear correlations	😐	☺	☺	😐	☺	☺	😐	☺	☺
	nonlinear correlations	😐	☹	☺	☹	☹	☺	☺	😐	☺
Speed	Training	☹	☺	☺	☺	☺	😐	☹	😐	☹
	Response	☺	☺	☹ / 😐	☺	☺	☺	😐	😐	😐
Robust -ness	Overtraining	☺	😐	😐	☺	☺	☹	☹	😐	😐
	Weak input variables	☺	☺	☹	☺	☺	😐	😐	😐	😐
Curse of dimensionality		☹	☺	☹	☺	☺	😐	☺	😐	😐
Transparency		☺	☺	😐	☺	☺	☹	☹	☹	☹

TMVA 4 – Preview on New Developments

- Data Regression
- Categorization: multi-class classification
- Automated classifier tuning: using cross-validation method
- Generic boost or bag of any classifiers
- Composite classifiers (parallel training in different phase space regions)
- Input data handling
  - Arbitrary combination of dataset transformations possible

Status: changed *T*MVA framework in handling datasets and classifiers, implemented regression training for most classifiers. User interface extended.

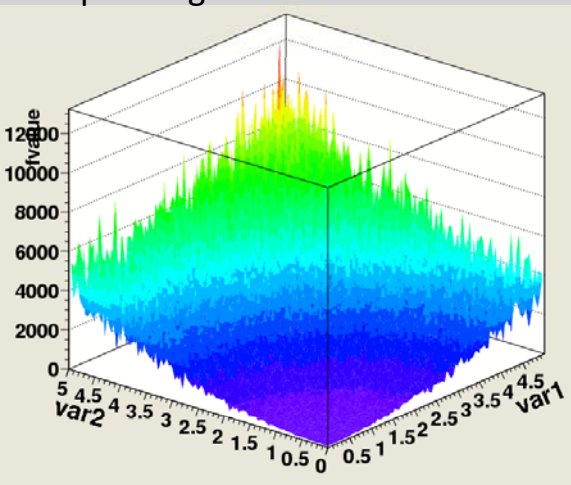
- New Method PDE Foam
  - Based on the ROOT TFoam by S.Jadach
    eprint physics/0203033
  - See next talk



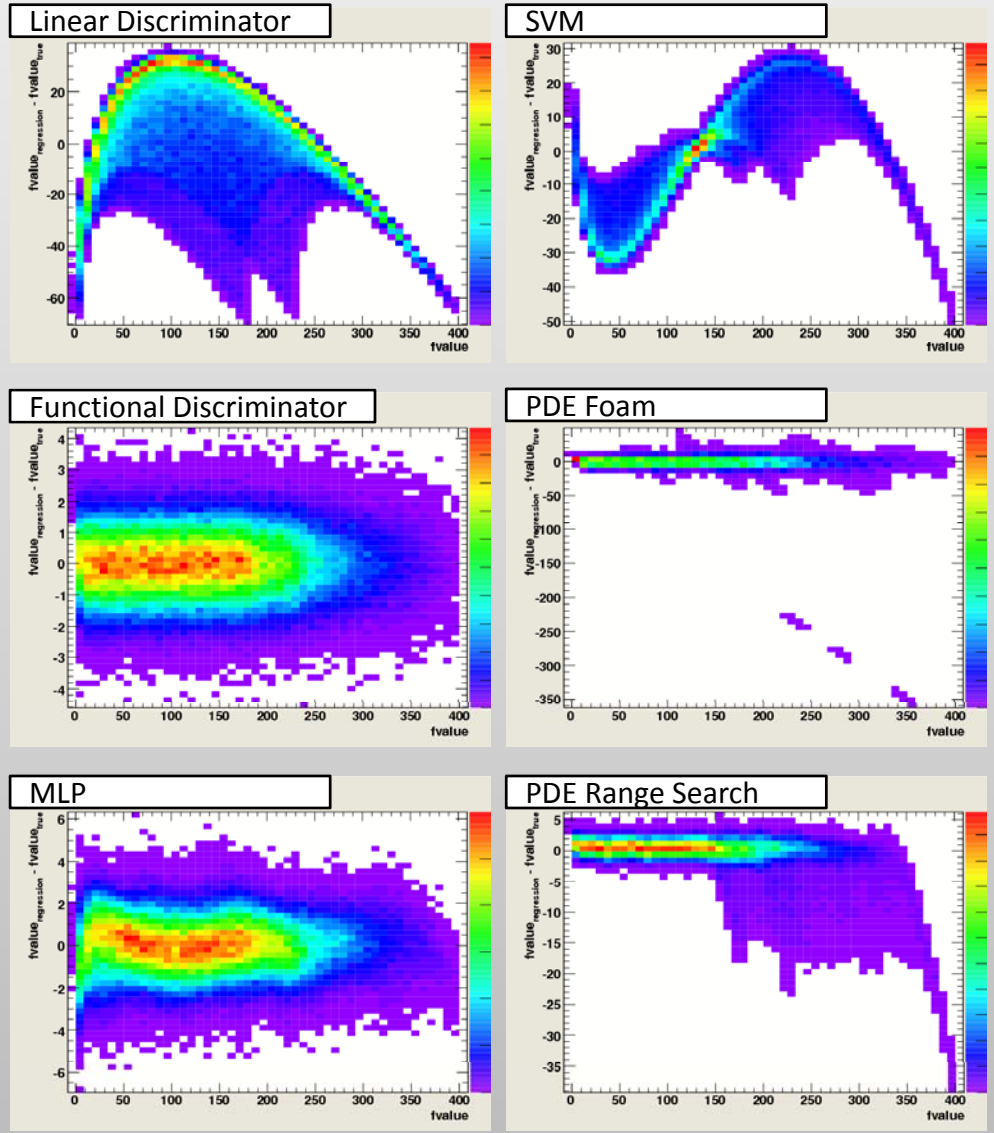
Multivariate Regression

- "Classifiers" try to describe the functional dependence
  - Example: predict the energy correction of jet clusters
- Classification: $R^N \rightarrow R \rightarrow \{0,1,..,N\}$
- Regression: $R^N \rightarrow R$
- Training: instead of specifying sig/bkgr, provide a regression target
  - Multi-dim target space possible
- Does not work for all methods!

Example: target as function of two variables



Δtarget vs. target on test sample for different "classifiers"



Automated Classifier Tuning

- Many classifiers have parameters that, being tuned, improve the performance
  - Overtraining: Performance on test sample worse than on training sample, classifier follows the particular features of the training sample to well
    - Protect by choosing right MLP training cycles, BDT splitting criteria
  - PDE range size, BDT pruning strength, Neural network structure

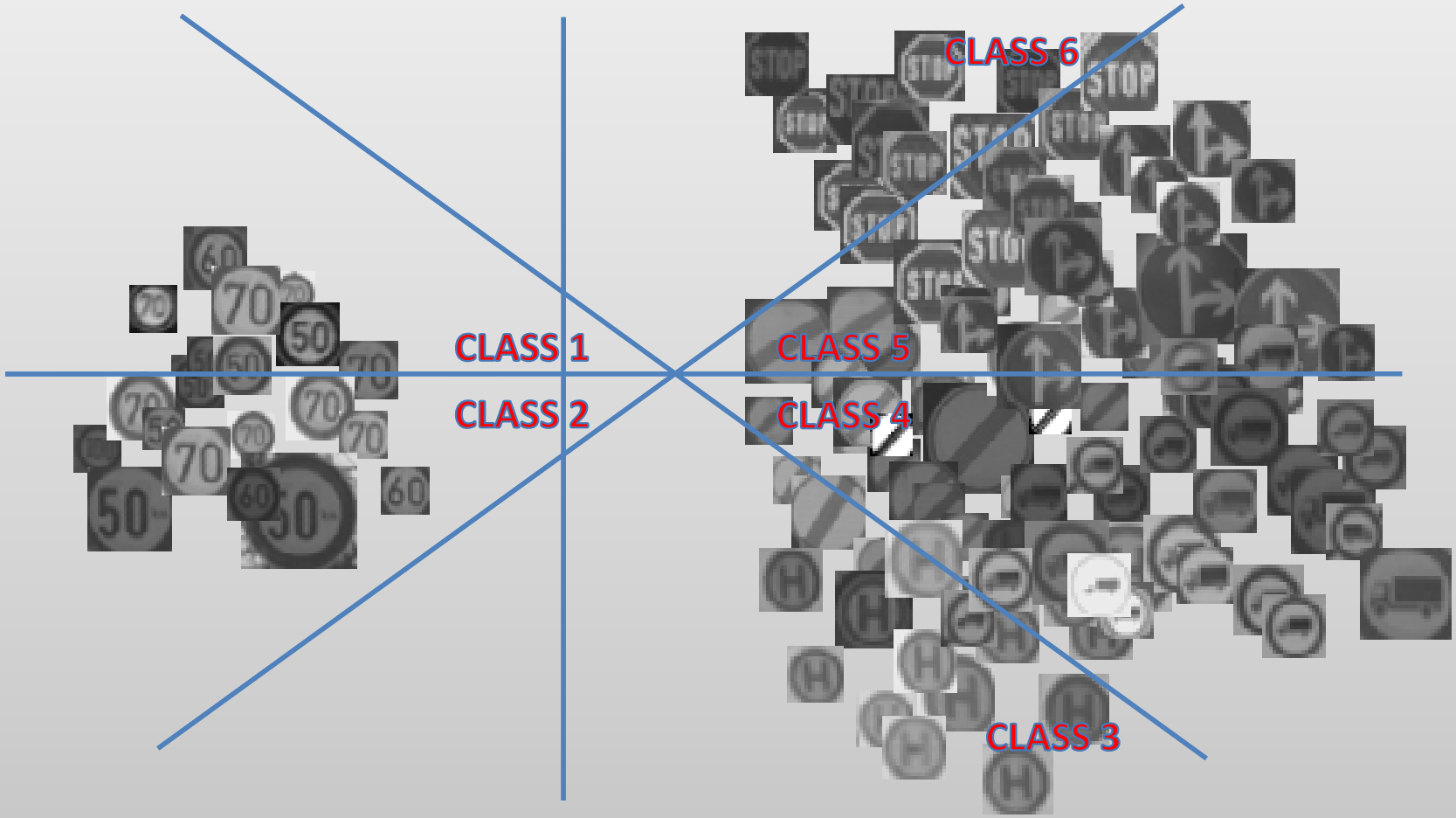
- Method for automated parameter tuning: Cross-Validation (aka Rotation-Estimation)

- Special choice of K-fold cross-validation:
  - Divide the data sample into K sub-sets
  - For set of parameters α train K classifiers $C_i(\alpha)$, i=1..K, omitting each time the i-th subset from the training to use as test sample

Test	Test	Test	Test	Test

  - Calculate test error $E_i(\alpha) = \langle \mathrm{L}(\mathrm{y}_i(x,\alpha)) \rangle_{\mathrm{Events}\ x}$ for each C_i and average $\overline{E}(\alpha)$
  - Choose tuning parameter set α for which $\overline{E}(\alpha)$ is minimum and train the final classifier using all data

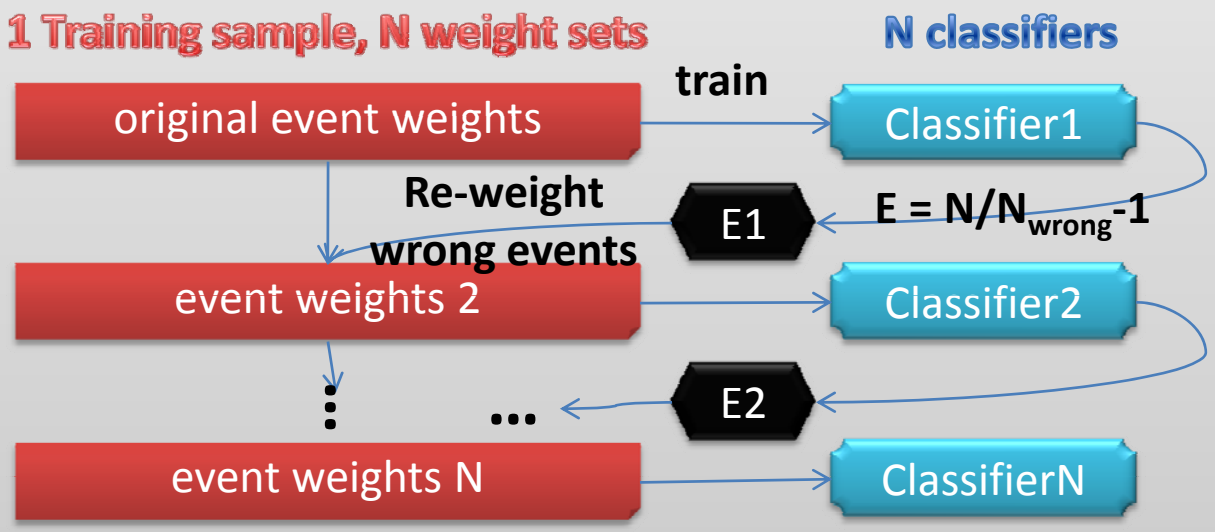
Multiclass Classification



- Some classifiers support this naturally: MLP, all Likelihood based approaches
- Others classifiers need special treatment: training 1 class against rest and make a class decision based on LH ratios which are based on the training output

Generic Classifier Boosting

- Principle: of multiple training cycles, each time wrongly classified events get a higher event weight



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

Response is weighted sum of each classifier response

- Tests should be performed with Cuts, MLP, and SVM
  - Likelihood based classifiers and Fisher can't be boosted

Further Plans

- TMVA to be made multi-threaded and run on multi-core architectures
  - Automatic tuning time consuming. Many users have multi-core machines and could take advantage of parallel computing
- Composite classifiers
  - Let's say you want to train a NN for tau selection differently in the detector barrel and endcap regions. The composite classifier can set this up and train automatically, need to specify the variable the region selector is based on (η)
  - Let's assume an N-class problem, and you want to use N Fisher classifiers one for each class to separate it from the rest. Then one could use a neural net on top to get a multi-class classifier
  - Just for convenience, can already be done manually
- Current stable TMVA version 3.9.5 for ROOT 5.22 (middle of December), afterwards moving to TMVA 4
  - Not everything at once:

  1) Regression and generic classifier boosting
  2) Multi-class classification, automatic classifier tuning
  3) Composite classifiers

  - Multi-core some time along the way

Summary

- Event selection in multi-dimensional parameter space should be done using multivariate techniques
  - Complex correlations between input variables need “machine treatment”
- Optimal event selection is based on the likelihood ratio
  - PDE RS, kNN, and Foam estimate the probability density in N dimensions but suffer from high dimensions
- Other classifiers with similar performance can be used in many cases:
  - Fishers Linear discriminant → simple and robust
  - Neural networks → very powerful but difficult to train (multiple local minima)
  - Support Vector machines → one global minimum but needs careful tuning
  - Boosted Decision Trees → “brute force method” with very good performance
- Interesting new features to come in TMVA 4
  - Regression will be very useful for a large variety of applications
  - Automatic tuning of parameters will improve quality of results, no need to be an expert