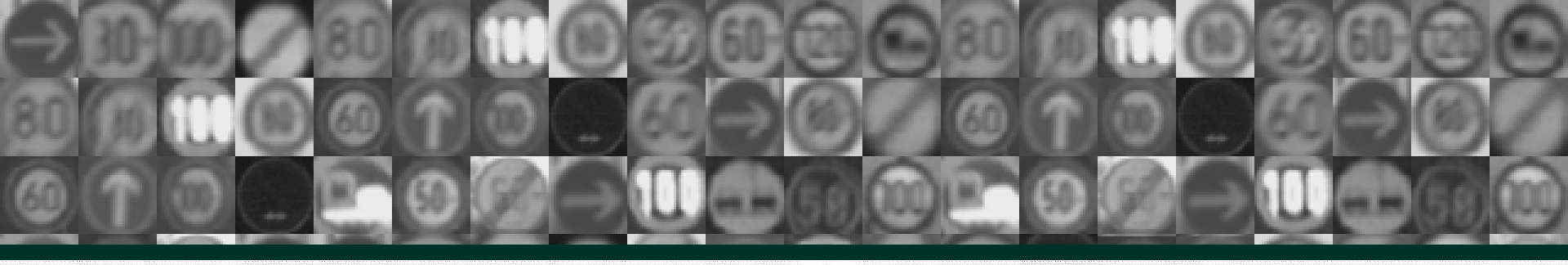


Data Analysis Techniques with *T*MVA

Jörg Stelzer* (DESY, Hamburg)

Presented by Dr. Attila Krasznahorkay (New York University)

13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research

Jaipur, India
February 22-27, 2010

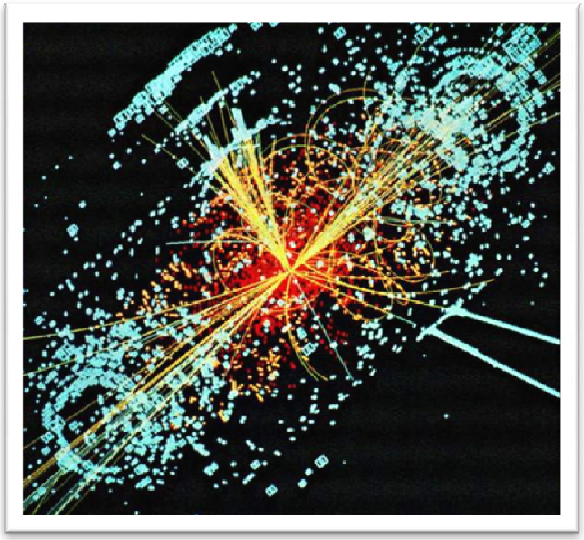
*For the TMVA developer team: A. Höcker, P Speckmayer, J.Stelzer, J. Therhaag, E. v. Törne, H.Voss and many contributors

Outline

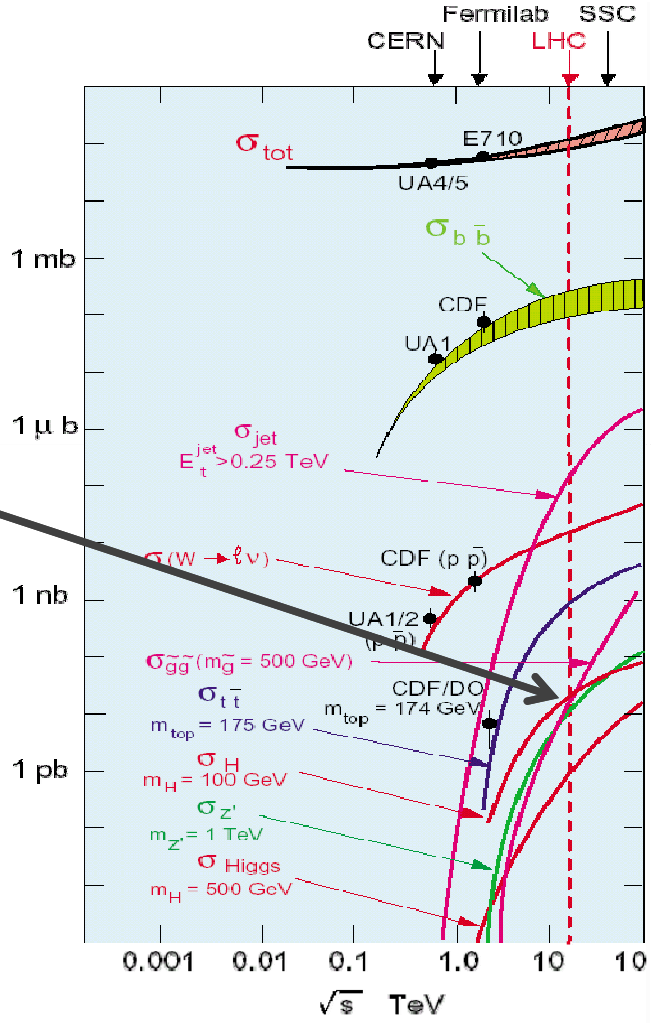
- Multivariate classification and regression and their application in High-Energy Physics
- TMVA – New Developments

Multi-Variate Techniques

MVA in High-Energy Physics (HEP)



A “typical” higgs event in the CMS detector, with ~25 minimum bias events underlying (simulated)



- ≡ Large datasets with small/tiny signal fractions.
- ≡ Increasingly important to use all the features of signal and background data (variable distributions, correlations)
- ≡ Complex data distributed in a high-dimensional space, difficult to disentangle without the help of automata.

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

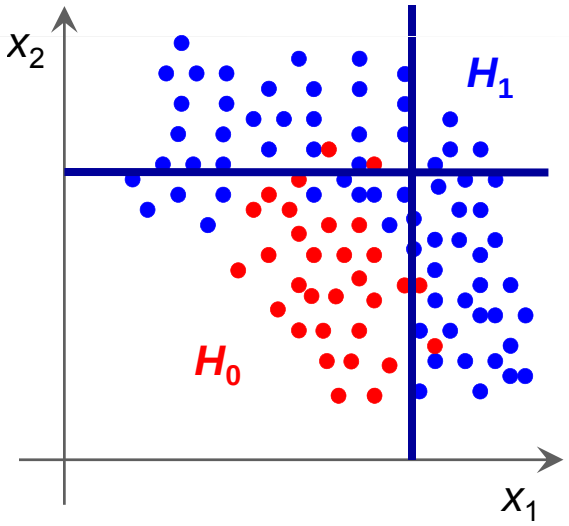
Examples of MVA in HEP

- Most HEP analyses require discrimination of signal from background:
  - Event level (Higgs searches, Top events, …)
  - Cone level (Tau-vs-jet reconstruction, …)
  - Track level (particle identification, …)
  - Lifetime and flavour tagging (b-tagging, …)
  - Parameter estimation (CP violation in B system, …)
  - etc.
- Regression analysis
  - Shower corrections (position, energy)
- 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)

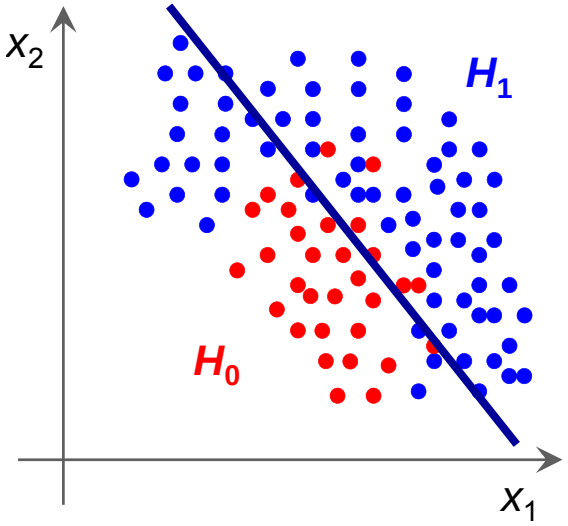
Classification

- 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 we use to select events of type H_1 ?

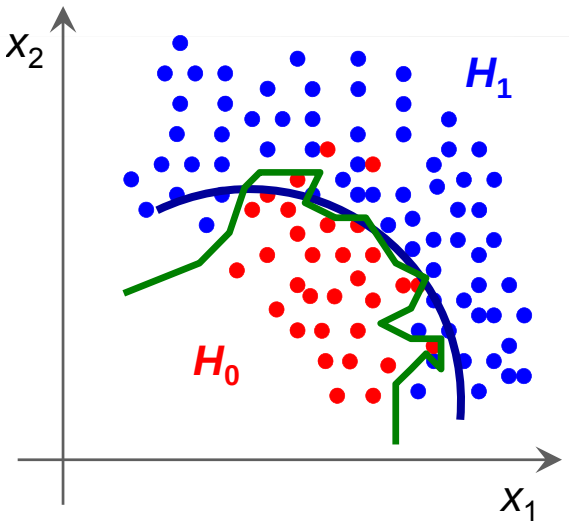
Rectangular cuts?



A linear boundary?

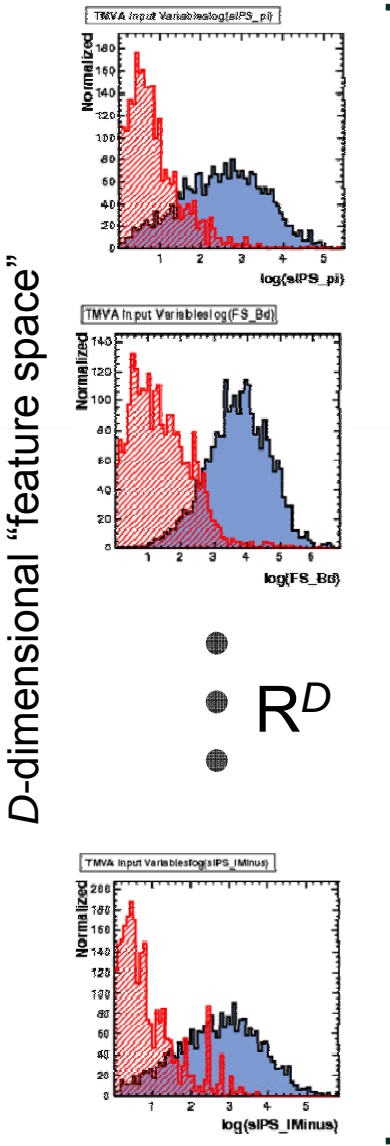


A nonlinear one?



How to decide this in an optimal way? ➔ Let the machine learn it !

Multivariate Classification



≡ Each event, Signal or Background, has *D* measured variables, lying in the *D*-dimensional "feature" space

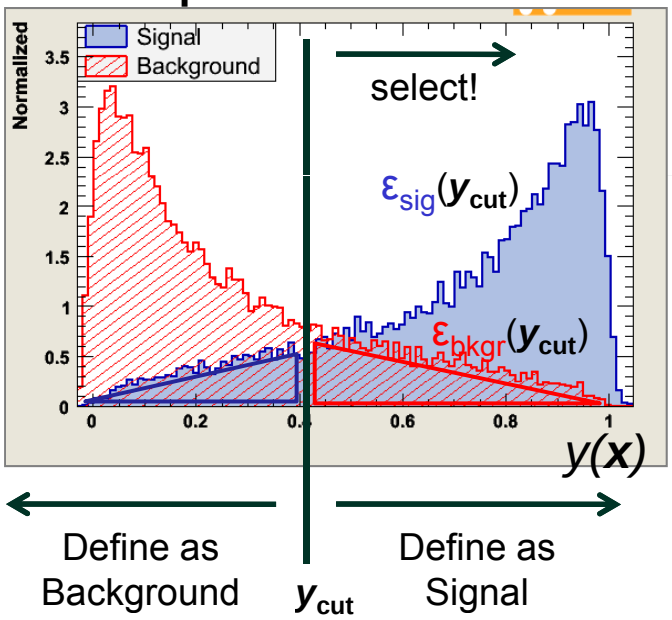
≡ Find a mapping from feature space to R (real number)

$y(\boldsymbol{x})$: $R^D \rightarrow R$

Analytic function y: $y(\boldsymbol{x}) = y_{cut}$ defining the decision boundary $(\boldsymbol{x})_{cut}$ in R^D

Overlap of signal and background distributions affects separation power and purity

MVA Output Distribution:



≡ Choosing y_{cut} defines size of type-1 and type-2 errors

Type-1 error: rate = ε_{bkgd}
classify event as signal even though it is not → loss of purity

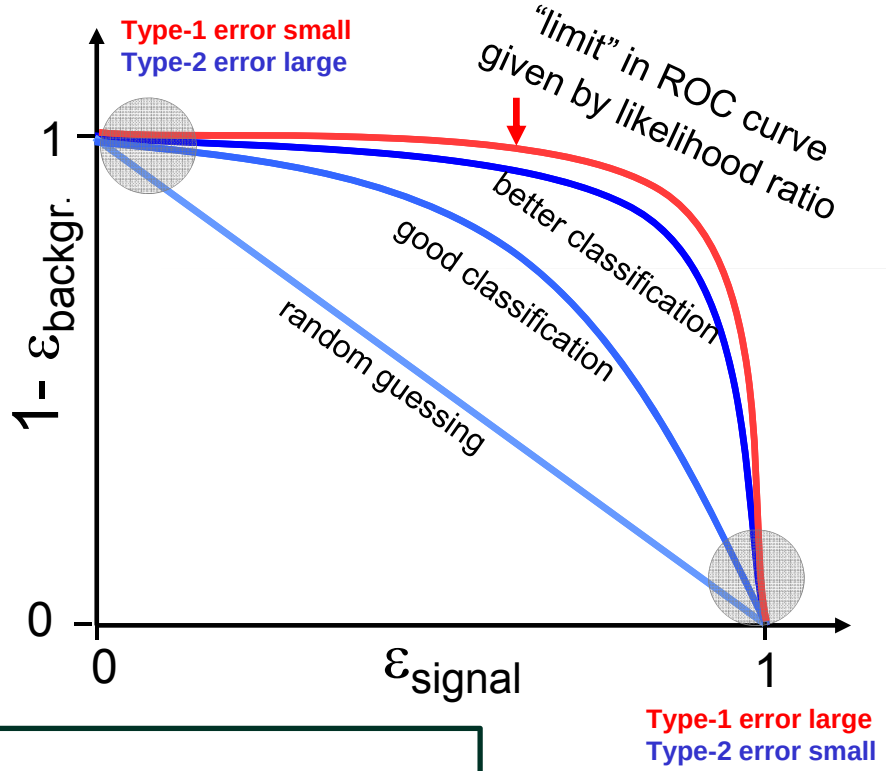
Type-2 error: rate = $1-\varepsilon_{sig}$
fail to identify signal as such → loss efficiency

Ideal Event Classification – Neyman-Pearson Lemma

≡ ROC (receiver operating characteristics) curve describes performance of a binary classifier by plotting the false positive (type-1 error) vs. the true positive fraction (signal efficiency)

≡ Likelihood ratio from the true probabilities of signal and background at point $\boldsymbol{x}$ in R^D

$$y(\boldsymbol{x}) = P(\boldsymbol{x}|S)/p(\boldsymbol{x}|B)$$



Neyman-Pearson:

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 ROC curve

Realistic Event Classification

- Unfortunately, the true probability densities functions $P(\boldsymbol{x}|S)$, $p(\boldsymbol{x}|B)$ are typically unknown
  - Neyman-Pearson's lemma doesn't really help us…

- Two ways out:
  - **Estimate PDFs** for signal and background from which an ***approximate likelihood ratio*** can be obtained
    → e.g. D-dimensional histogram, Kernel density estimators, …
  - Find a "**discrimination function**" and corresponding ***decision boundary in feature space***, i.e. a hyper-plane, that separates signal and background
    → e.g. Linear Discriminator, Neural Networks, Support Vector Machines, …

To achieve this *T*MVA uses supervised learning

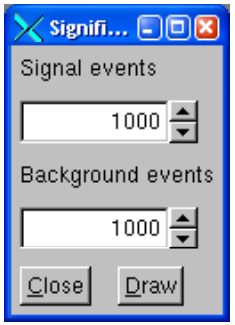
- **Supervised**: use of ***known events*** (already classified) for training
  - Typically: Monte Carlo data, cleanly identified data sample

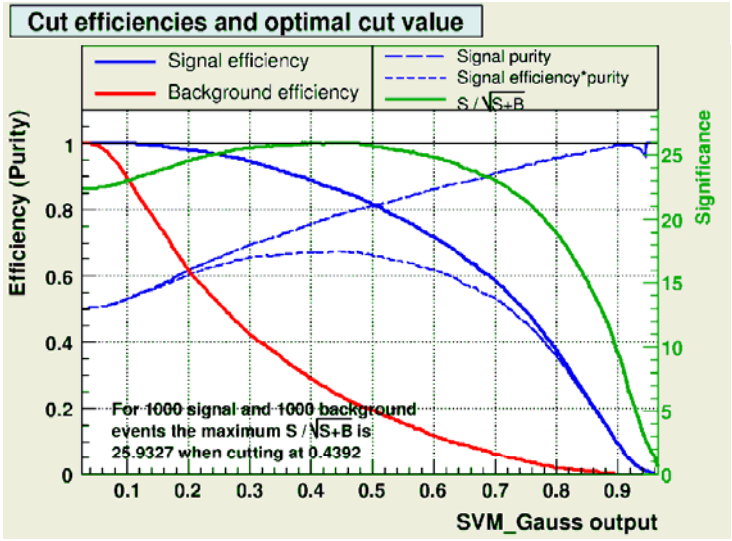
From the ROC Curve to the Signal Selection

- Cut on classifier output (= point on ROC curve): ε_{sig}, ε_{bkgd}
  - Together with size of underlying data sample (N_{sig}, N_{bkgd}) gives number of selected signal and background events: $S = \varepsilon_{sig} N_{sig}$, $B = \varepsilon_{bkgd} N_{bkgd}$

- Optimal cut, the working point, depends on the problem:
  - Cross section measurement: maximum of $S/\sqrt{(S+B)}$
  - Search: maximum of $S/\sqrt{(B)}$
  - Precision measurement: high purity
  - Trigger selection: high efficiency

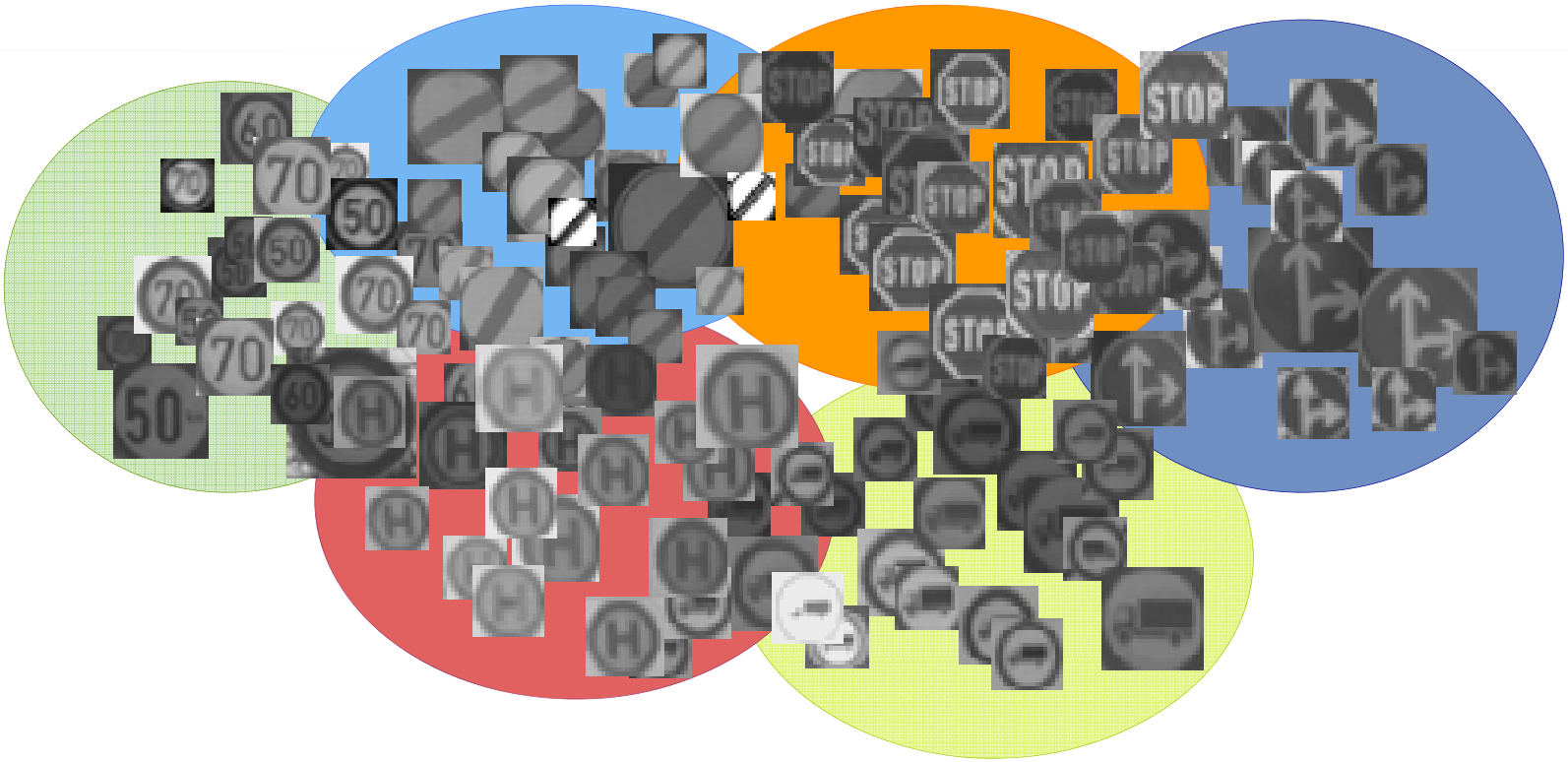
- and the expected number of events N_{sig}, N_{bkgd}:





Multi-Class Classification

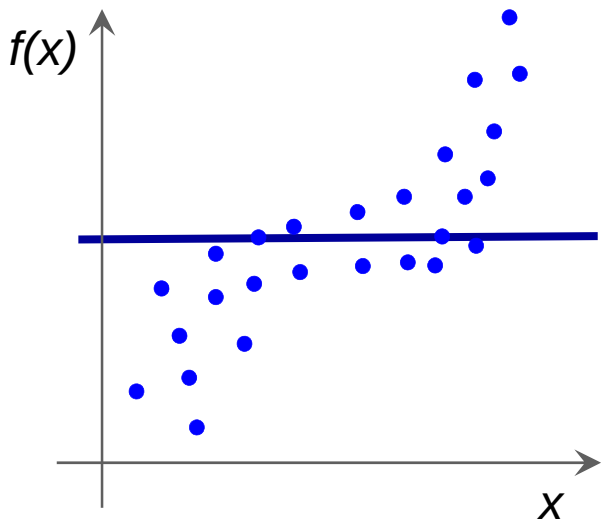
- No ROC curve, classifier outputs class index
  - Training goal: maximize predictive accuracy by **minimizing misclassification cost** (usually = misclassification rate)
  - Most criteria, e.g. misclassification rate, depend on size of class samples → training sample size (or weight) needs to reflect expected data



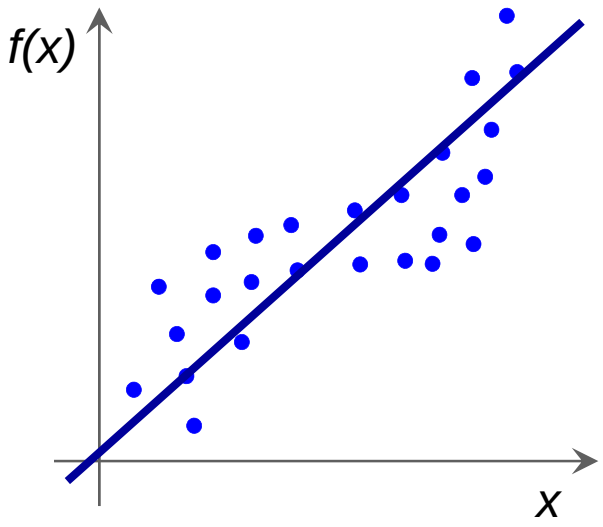
Regression

- Another multivariate problem: estimation of the "functional behaviour" from a set of measurements?
  - Energy deposit in a the calorimeter, distance between overlapping photons, …
  - Entry location of the particle in the calorimeter or on a silicon pad, …

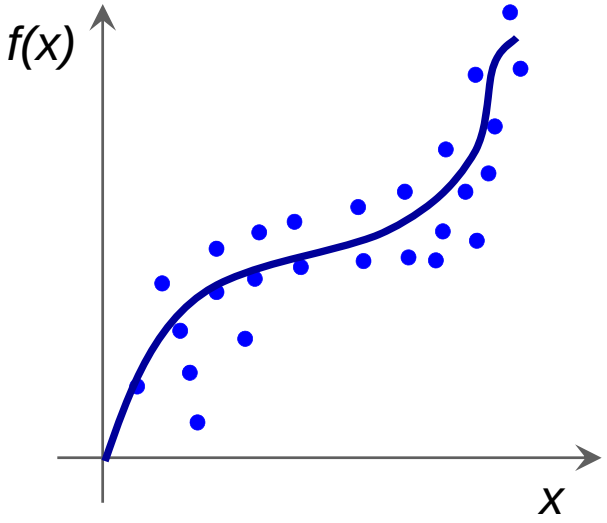
Constant ?



Linear function ?



Nonlinear ?



Seems trivial? What if we have many input variables? ➔ Machine learning!

TMVA

Recapture

- ***TMVA*** implements a large number of state-of-the-art classifiers
  - Conventional linear and non-linear classifiers (likelihood-based, discriminators), modern classifiers like Foam boxing PDE, Boosted DTs, Support Vector Machines, Rule ensembles
- TMVA approach:
  - Have one common platform / interface for all MVA classifiers
  - Have common data pre-processing capabilities
  - Provide common data input and analysis framework
  - 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
  - Many presentations about usage of TMVA, collected at http://tmva.sourceforge.net/talks.shtml
- TMVA @ ACAT 2008
  - http://indico.cern.ch/contributionDisplay.py?contribId=2&sessionId=5&confId=34666
  - Idea behind TMVA, overview and comparison of the methods, outlook to new developments
- Today: Update on recent developments in TMVA

New Developments in TMVA

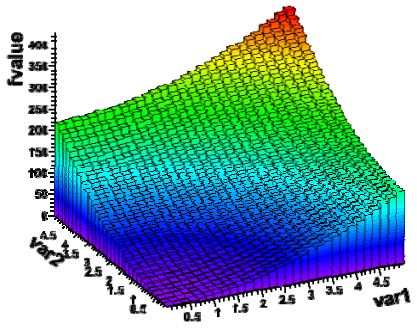
- TMVA 4 first released with ROOT 5.24. New Features:
  - Multivariate multi-target regression
  - Generic boosting
  - Category classifier (TMVA 4.0.4 in ROOT 5.26)
  - New methods: PDEFoam and Bayesian Neural Net
  - Framework changes: chained data preprocessing, weight-files in xml format (text format can still be read), internal reorganization to prepare for composite classifiers
- TMVA development moved to ROOT SVN repository
  - TMVA releases more tightly coupled to ROOT releases, following ROOT strategy of development, production, and patch releases
  - Still post all TMVA releases to sourceforge. Standalone, and compatible with older ROOT releases.
- Two new core developers: J. Therhaag, E. v. Toerne

Regression

Multivariate (Multi-target) Regression

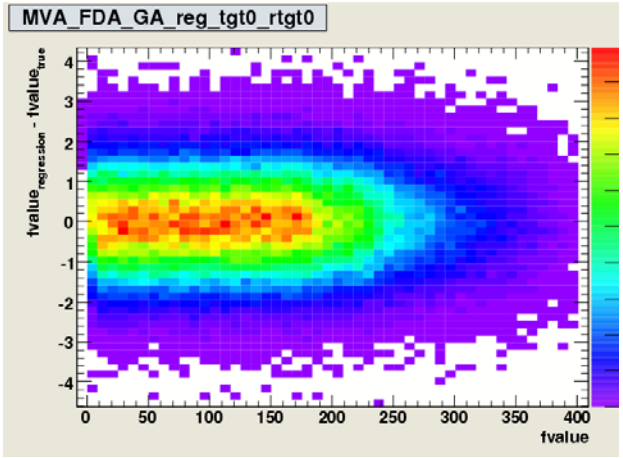
- Regression approximates the functional dependence of a target on $(x_1,\ldots,x_N)$
  - Example: predict the energy correction of jet clusters in calorimeter
- Training: instead of specifying class type (sign/bkgr), provide a regression target
- Not yet implemented for all methods (so far: LD, PDERS, PDEFoam, MLP, SVM, GradBoost-BDT)
  - Multi-dim target space possible (framework implemented, a few classifiers already adapted)
- Regression performance evaluated on test sample by comparing target prediction with target

Evaluate: deviation of target prediction from target …

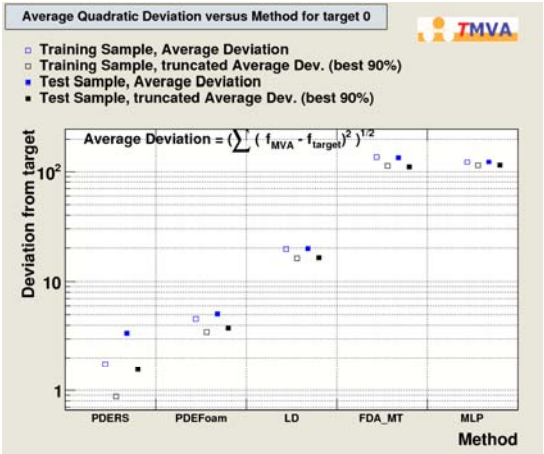


Example: Target as function of 2 input variables

… as a function of target:



… averaged over test sample:

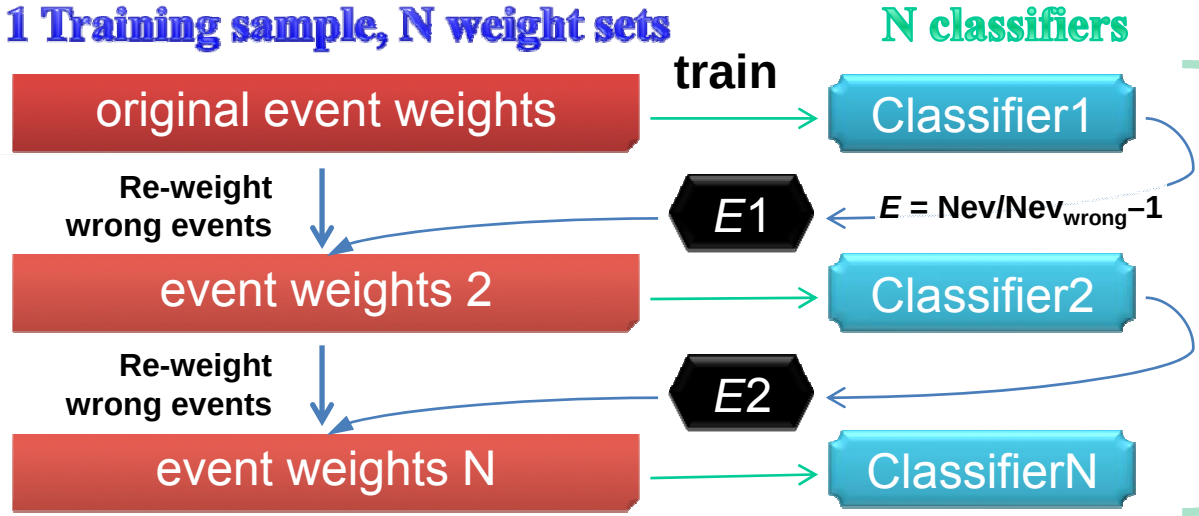


Combining Classifers

- Generalised Boosting of any classifier
- Categorising classifiers

Generalised Classifier Boosting

≡ Principle (just as in BDT): 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

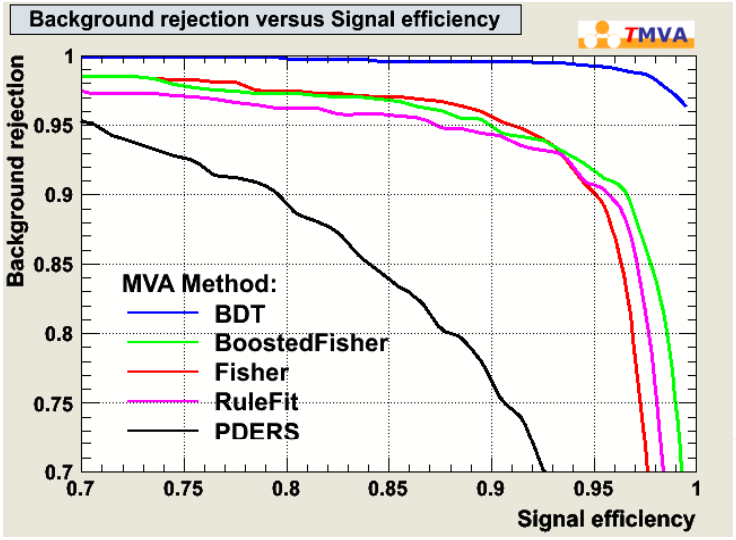
≡ Boosting will be interesting especially for Methods like Cuts, MLP, and SVM, but also Fisher

Example – Boosted Fisher

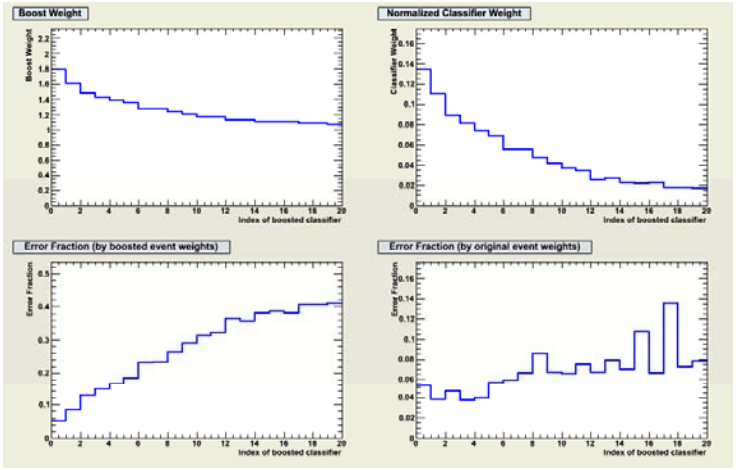
- Sample of traffic signs (16x16 pixel png-format)
  - Large number of input variables (256) with values between 1 and 255
  - Sparse population of input-space (PDERS suffers from large dimensionality)

Signal

Background



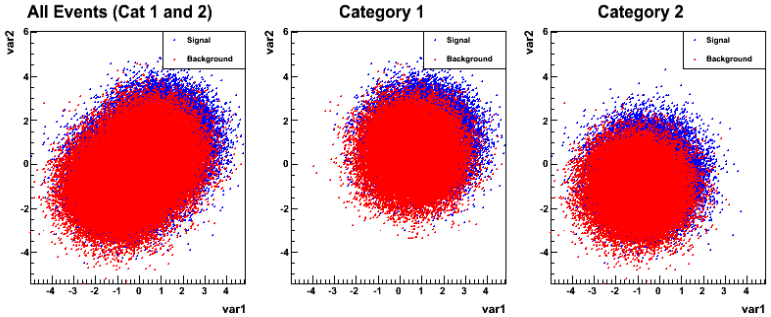
Boost evolution



- Boosting Fisher shows improved performance over simple Fisher
  - Following the boost evolution: errors of boosted trees grow → smaller contribution than original

Categorising Classifiers

- Multivariate training samples often have ***distinct sub-populations***
  - A detector element may only exist in the barrel, but not in the endcaps
  - A variable may have different distributions in barrel, overlap, and endcap regions



Toy example: correlation between variables in full sample disappears not present within sub-populations

- Ignoring this dependence ***creates correlations between variables***, which must be learned by the classifier
  - Classifiers such as the projective likelihood, which do not account for correlations, significantly loose performance if the sub-populations are not separate

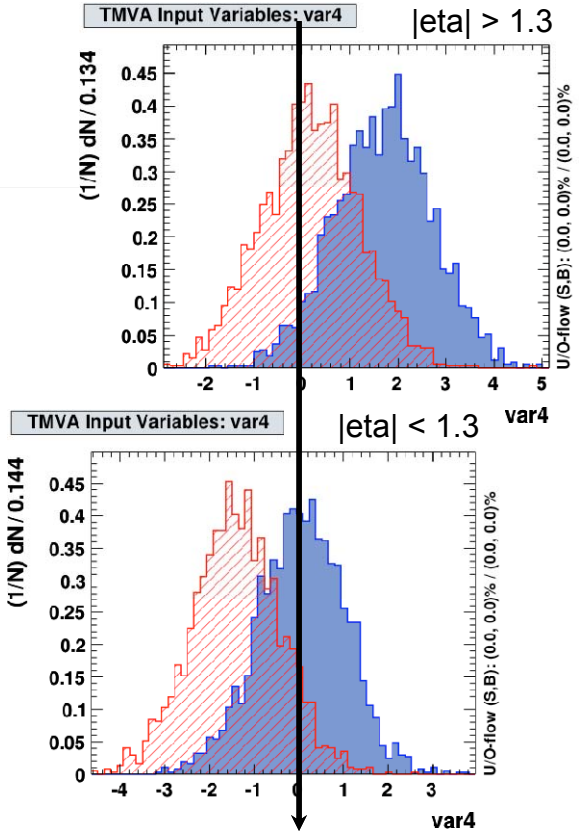
- Categorisation means splitting the data sample into categories defining disjoint data samples with the following (idealised) properties:
  - Events belonging to the same category are statistically **indistinguishable**
  - Events belonging to **different categories have different properties**

- Categories are treated independently for training and application (transparent for user), but evaluation is done for the whole data sample (comparison to others)

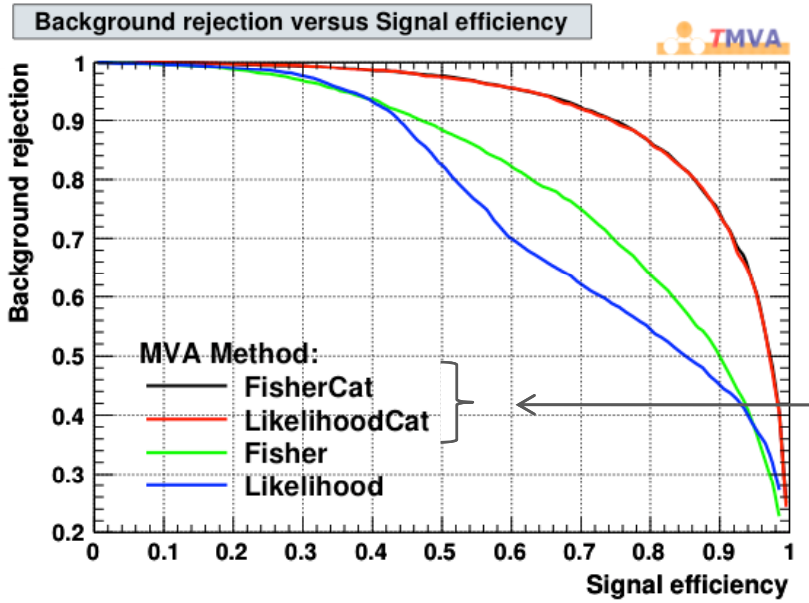
Categorising Classifiers – An Example

Example of 4 Gaussian-distributed input variables:

- "var4" depends on a variable "eta" ("eta" not to be used for classification): using the category classifier with two categories (|eta| < 1.3, |eta| > 1.3) improves performance significantly



Signal and Background Gaussian means are shifted between the two categories



Recover optimal performance after splitting into categories

The category technique is heavily used in multivariate likelihood fits, e.g., RooFit (RooSimultaneousPdf)

New Classifiers

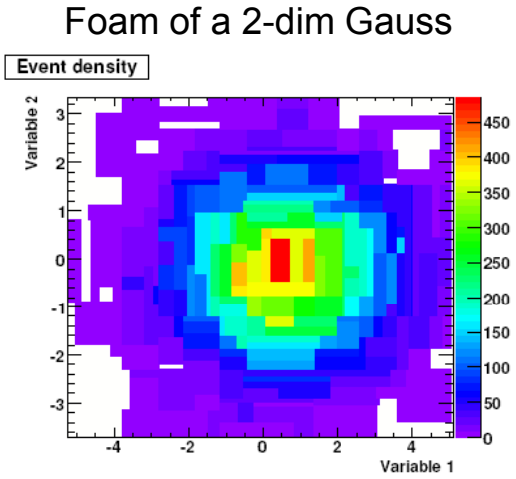
- PDEFoam
- Bayesian Neural Net

New Classifiers

PDEFoam

≡ Multi-dimensional likelihood based method (PDE)

- = Extends PDE-RangeSearch
- = Uses Foam* technique for sampling the input parameter space (self adaptive binning with minimized cell variance)
- = Similar performance to PDERS, significant speed improvement for application
- = Implementation by Alexander Voigt



* S. Jaddach, "Foam: A General-Purpose Cellular Monte Carlo Event Generator", CERN-TH/2002-059, physics/0203033

Bayesian NN

≡ Emphasis on probability interpretation of prediction

- = Refined selection of cost function
- = Use regulator to reduce complexity. Dragging unimportant weights to 0 → overtraining prevention
- = Implementation by Jiahang Zhong

Wrap Up

Multivariate Analysis in HEP

- The HEP community has already a lot of experience with MVA classification
  - In particular for rare decay searches, almost all mature experiments use it
  - They increase the experiment's sensitivity, and may reduce systematic errors due to a smaller background component
  - MVAs are not black boxes, but (possibly involved) $R^D \rightarrow R$ mapping functions

- Need to acquire more experience in HEP with multivariate regression
  - Our calibration schemes are often still quite simple: linear or simple functions, look-up-table, mostly depending on few variables (e.g., η, p_T)
  - Non-linear multivariate regression may significantly boost calibration and corrections applied, in particular if it is possible to train from data

Functionality in Preparation

≡ Finish multiclass classification

- = Framework implemented. Some classifiers already extended to work for multiple classes (MLP, FDA, BDTGradBoost)

≡ Cross-Validation

- = Overtraining protection
- = Automated classifier tuning

≡ Combination of classifiers

- = E.g., simple classifiers can first combine uncorrelated variables, their output can be fed into other classifiers → performance improvement

At the End a Fun Example

≡ Watching the Training Progress of a MLP