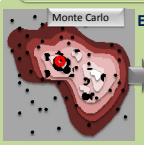
Multivariate Data Analysis with TMVA 4

TMVA core developer team: A. Höcker, P.Speckmayer, J.Stelzer, H.Voss; ***arXiv physics/0703039***

The toolkit for multivariate analysis, TMVA, provides a large set of advanced multivariate analysis techniques for signal/background classification. In addition, TMVA now also contains regression analysis, all embedded in a framework capable of handling the pre-processing of the data and the evaluation of the output, thus allowing a simple and convenient use of multivariate techniques. The analysis techniques implemented in TMVA can be invoked easily and the direct comparison of their performance allows the user to choose the most appropriate for a particular data analysis.

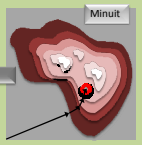
minimization



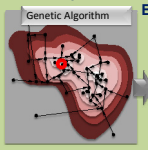
Brute force method

- Sample entire solution space, and chose solution providing minimum estimator
- Good global minimum finder, but poor accuracy

Default solution in HEP: Minuit



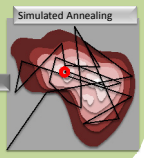
- Gradient-driven search, using variable metric, can use quadratic Newton-type solution
- Poor global minimum finder, gets quickly stuck in presence of local minima



Biology-inspired

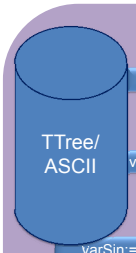
- "Genetic" representation of points in the parameter space
- Uses mutation and "crossover"
- Finds approximately global minima

Like heating up metal and slowly cooling it down ("annealing")



Atoms in metal move towards the state of lowest energy while for sudden cooling atoms tend to freeze in intermediate higher energy states → slow "cooling" of system to avoid "freezing" in local solution

preprocessing

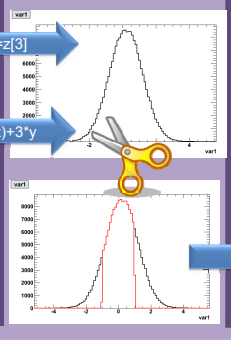


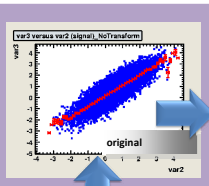
Data input

- Combination or function of available variables
- Similar to ROOT's TTree::Draw()

Apply selection cuts

- Possible independently for each class



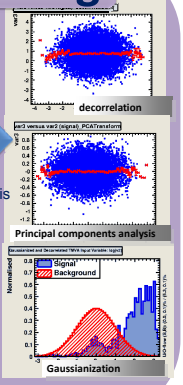


Define event weights

- Can be defined globally, tree-wise and class-wise

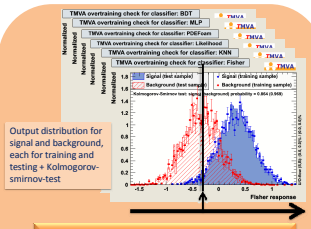
Transformations

- Can be set for each Method independently
- TMVA knows:
  - Normalisation
  - Decorrelation
  - Principal component analysis
  - Gaussianisation



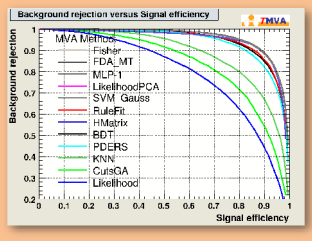
- Transformations can be chained
- Transformations improve result in case of projective Likelihood and linear correlations of variables
- May not improve classification in case of strong non-linear correlations

classification

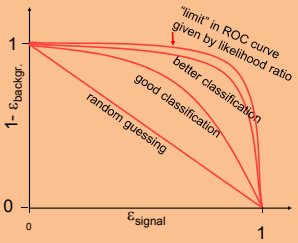


Output distribution for signal and background, each for training and testing + Kolmogorov-smirnov-test

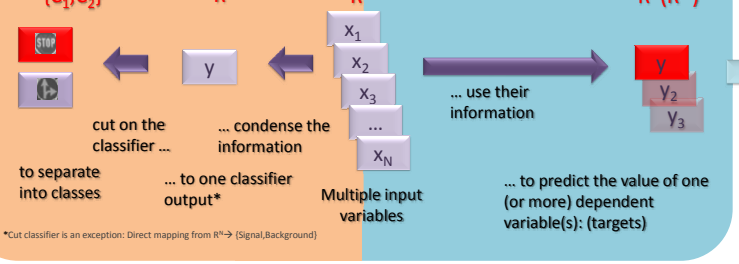
Scan over classifier output variable creates set of (ε_{sig},1-ε_{bkgd}) points = ROC curves



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



$\{C_1,C_2\}$ ← R ← R^N → R (R^M)



cut on the classifier ... to separate into classes

... condense the information ... to one classifier output*

Multiple input variables

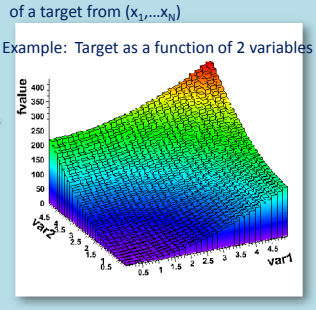
... use their information ... to predict the value of one (or more) dependent variable(s): (targets)

*Cut classifier is an exception: Direct mapping from R^N→ {Signal,Background}

regression

Approximates the functional dependence of a target from (x_1,...x_N)

Example: Target as a function of 2 variables



evaluation

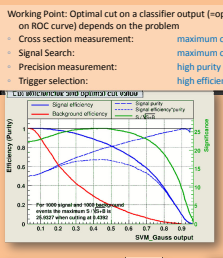
False positive (type 1 error) = ε_{backgr}: classify background event as signal → loss of purity

False negative (type 2 error) = 1-ε_{signal}: fail to identify a signal event as such →loss of efficiency

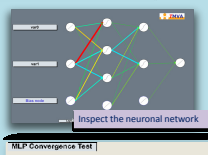
TMVA provides many evaluation macros producing plots and numbers which help the user to decide on the best classifier for an analysis

Working Point: Optimal cut on a classifier output (=optimal point on ROC curve) depends on the problem

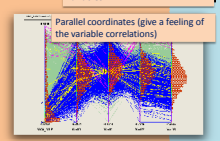
Cross section measurement:	maximum of S/√(S+B)
Signal Search:	maximum of S/√(B)
Precision measurement:	high purity
Trigger selection:	high efficiency



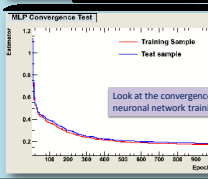
Correlation Matrices for the input variables



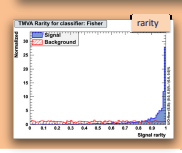
Inspect the neuronal network

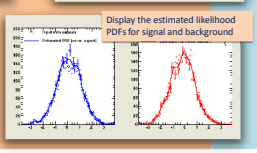


Parallel coordinates (give a feeling of the variable correlations)



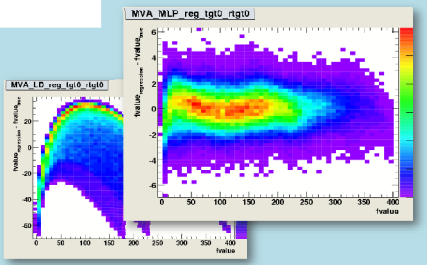
Look at the convergence of the neuronal network training



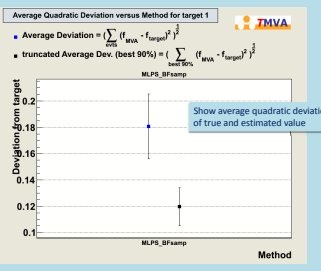


Display the estimated likelihood PDFs for signal and background

Inspect the BDT



Estimated value minus true value as a function of the true value



Show average quadratic deviation of true and estimated value

summary

Classification	Regression	
		Conventional linear methods (Should only be used if non-linear correlations are absent)
☑	☒	Cut based (still widely used since transparent)
☑	☑	Projective likelihood estimator (optimal if no correlations)
☑	☑	Linear Fisher Description (robust and fast)
		Common non-linear methods
☑	☑	Neural Network (powerful, but challenging for strongly non-linear feature-space)
☑	☑	PDE: Range Search, kNN, Foam (multi-dim LHood → optimal classification)
☑	☑	Functional Description Analysis
		Modern methods recent in HEP
☑	☑	Boosted Decision Tree (brute force, not much tuning necessary)
☑	☑	Support Vector Machine (one global minimum, careful tuning necessary)
☑	☒	Learning via rule ensembles

- Large number of MVA methods implemented
- One common platform/interface for all MVA methods
- Common data pre-processing capabilities
- Common input and analysis framework (ROOT scripts)
- Train and test all methods on same data sample and evaluate consistently
- Method application w/ and w/o ROOT, through macros, C++ executables or python

Criteria		Cuts	Likelihood	PDERS/ k-NN	H-Matrix	Fisher	MLP	BDT	RuleFit	SVM
Performance	no / linear correlations	😐	😊	😊	😐	😊	😊	😐	😊	😊
	nonlinear correlations	😐	☹	😊	☹	☹	😊	😊	😐	😊
Speed	Training	☹	😊	😊	😊	😊	😐	☹	😐	☹
	Response	😊	😊	☹ 😐	😊	😊	😊	😐	😐	😐
Robustness	Overtraining	😊	😐	😐	😊	😊	☹	☹	😐	😐
	Weak input variables	😊	😊	☹	😊	😊	😐	😐	😐	😐
Curse of dimensionality		☹	😊	☹	😊	😊	😐	😊	😐	😐
Transparency		😊	😊	😐	😊	😊	☹	☹	☹	☹