Analisi Statistiche Multivariate con il pacchetto TMVA

Università degli studi di Padova

21 Gennaio 2014

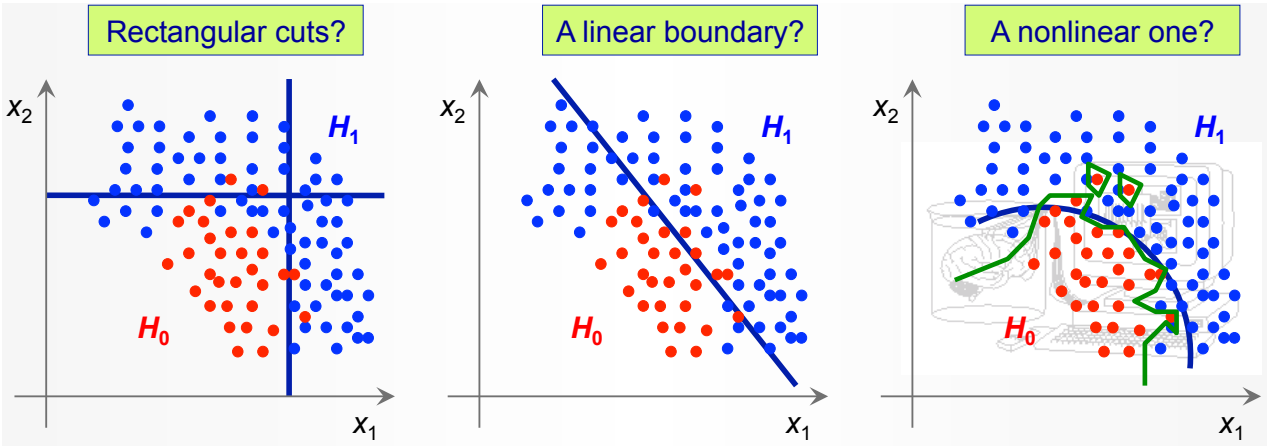
Credits

Il materiale utilizzato per la presentazione proviene da:

- Andreas Hoecker (CERN), "Multivariate Data Analysis Techniques", presentazione al Workshop on Statistics, Karlsruhe, Germany, 12-14 Ottobre 2009
- Helge Voss (MPI-K, Heidelberg), "Data Analysis with TMVA", Seminario a Lausanne, 12 Aprile 2010
- A. Hoecker et al, "TMVA - Toolkit for Multivariate Data Analysis", User Guide, arXiv:physics/0703039

Classificazione di Eventi

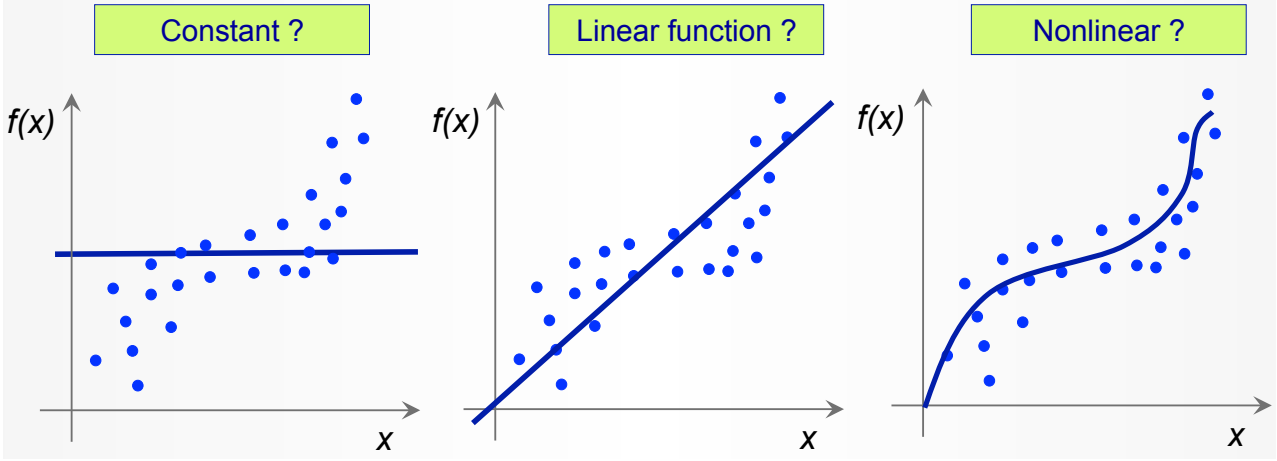
- La separazione di segnale dal fondo è uno dei tipici problemi che si presentano nell'analisi dei dati sperimentali
- Supponiamo di avere due categorie di eventi H_0 e H_1
- Si sono individuate osservabili che permettono la discriminazione x_1, X_2, . . ., x_n
- Studiamo dei "tagli" nello spazio dei parametri per separarli



- Qual'è il criterio migliore per isolare il segnale ?

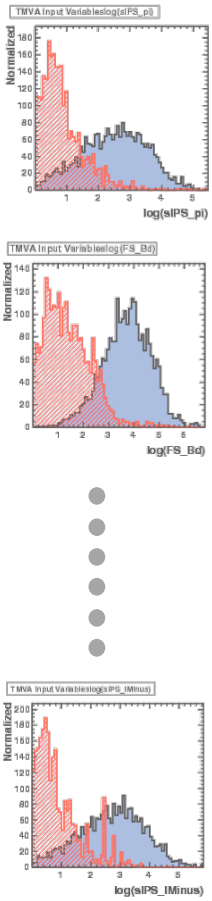
Regressione

- Un altro problema ricorrente è la ricerca di un modello funzionale che descriva i dati



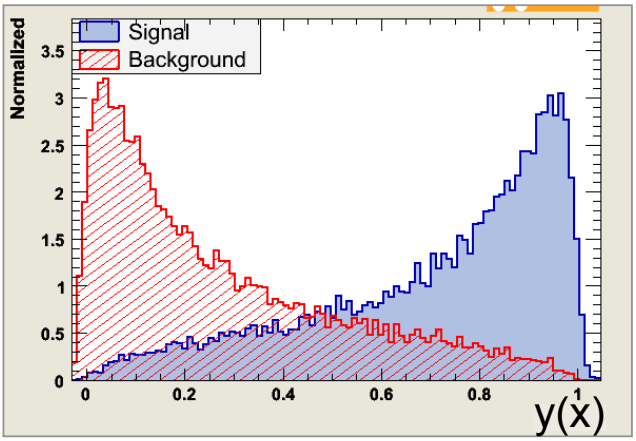
- Il problema è tutt'altro che banale specialmente se lo spazio degli osservabili è multidimensionale

La classificazione degli eventi in N-dim

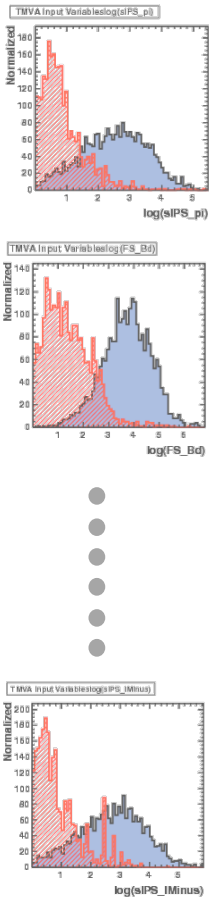


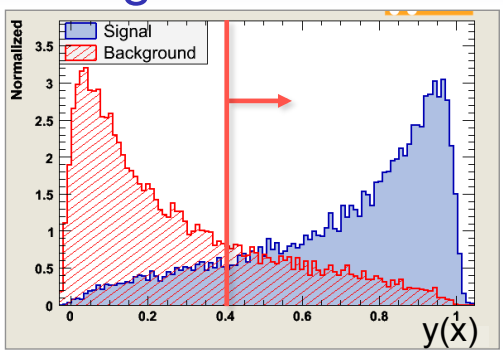
- Per ogni evento (segnale o background) vengono misurate N variabili
- Si cerca un mapping dallo spazio dei parametri a N-dimensioni ad una variabile di output

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



La classificazione degli eventi in N-dim





- $y(x)$ rappresenta una variabile statistica di test nello spazio a N-dim dei parametri
- Le distribuzioni di y, $\mathrm{PDF}_{signal}(y)$ e $\mathrm{PDF}_{background}(y)$ vengono utilizzate per definire tagli di selezione
- I parametri importanti per decidere il taglio sono:
  - efficienza : frazione di eventi di segnale accettato sul totale degli eventi di segnale $N_s(y > cut)/N_s$
  - purezza : frazione di eventi segnali presenti nel campione selezionato $N_s(y > cut)/N(y > cut)$

Classificazione di Eventi

- $P(classe = C|x)$ è la probabilità che un evento sia di classe C, date un insieme di osservabili misurate $\vec{x} = \{x_1, x_2, \ldots, x_N\}$

Densità di Probabilità dipendente dalle misure $\vec{x}$ e dalla funzione di mapping

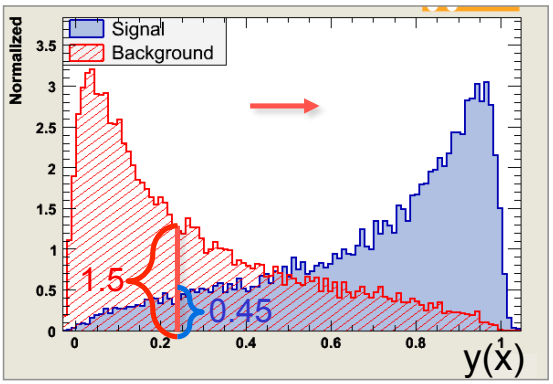
Probabilità Priori di osservare un evento di classe-C (equivale alla frazione di eventi di segnale sul totale signal+background)

$$P(classe = C|y) = \frac{P(y|C)P(C)}{P(y)}$$

Probabilità a posteriori

Densità di probabilità globale di effettuare la misura y(x)
$P(y) = \sum_{classi} P(y|Classe)P(Classe)$

Esempio di classificazione



- $\text{PDF}_{background}(y)$ e $\text{PDF}_{signal}(y)$ rappresentano le distribuzioni di probabilità normalizzate per gli eventi di fondo e di segnale
- Supponiamo di avere un evento sconosciuto per il quale si misura $y(x) = 0.2$

- calcoliamo $\text{PDF}_{background}(y) = 1.5$ e $\text{PDF}_{signal}(y) = 0.45$
- supponendo che f_S e f_B siano le frazioni si segnale e di fondo nel campione, possiamo scrivere

$$P(signal|y) = \frac{f_S PDF_{signal}(y)}{f_S PDF_{signal}(y) + f_B PDF_{background}(y)}$$

- $P(signal|y)$ è la probabilità che un evento sia di segnale, dati gli osservabili misurati $\vec{x} = \{x_1, x_2, \ldots, n_N\}$

Bayes Optimal Classification

$$P(classe = C|y) = \frac{P(y|C)P(C)}{P(y)}$$

- Per separare il Segnale rispetto al Background, si sceglie come criterio di selezione

Posterior odd ratio

$$\frac{P(S|y)}{P(B|y)} = \frac{P(y|S)}{P(y|B)} \cdot \frac{P(S)}{P(B)} > c$$

$c \propto$ efficienza e purezza

Likelihood ratio come funzione discriminante in y(x)

Prior odds ratio di selezionare un evento di segnale (probabilità relativa del segnale versus fondo)

Errori di classificazione

- Nella selezione di eventi di Segnale su eventi di Background si può incorrere in
- **errori di I tipo** :
  - si classifica un evento di classe C ma non lo è:
    (si rigetta l'ipotesi nulla sebbene sarebbe stata quella corretta)
  - → perdita di purezza
- **errori di II tipo** :
  - non si classifica un evento che sarebbe stato di classe C:
    (si rigetta un'ipotesi che sarebbe stata quella vera)
  - → perdita di efficienza

Trying to select signal events:
(i.e. try to disprove the null-hypothesis stating it were "only" a background event)

accept as: / truly is:	Signal	Back-ground
Signal	☺	Type-2 error
Back-ground	Type-1 error	☺

- se $A =$ regione nella quale si accettano gli eventi come segnale:
- $\alpha = \int_A P(x|B)dx$, $\alpha =$ background selection efficiency
- $\beta = \int_{\overline{A}} P(x|S)dx$, $1 - \beta =$ signal selection efficiency

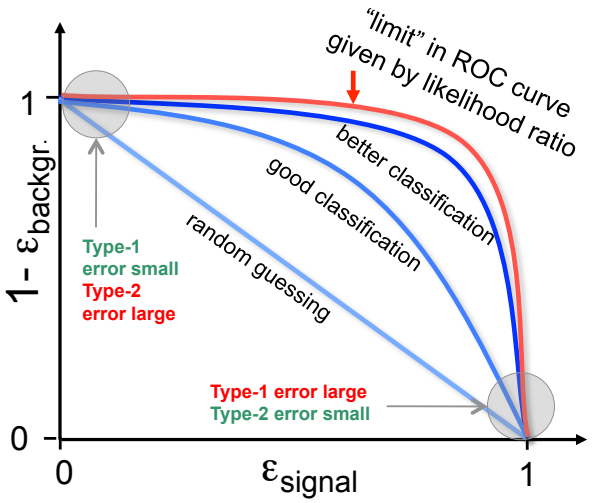
Lemma di Neyman-Pearson

- Rapporto di verosimiglianza (Likelihood):

$$y(x) = \frac{P(x|S)}{P(x|B)}$$

- **Neyman-Pearson** :

Il rapporto di verosimiglianza usato come criterio di selezione fornisce per ogni efficienza di selezione la migliore reiezione di background possibile



- Spostando il taglio si $y(x)$ si muove il punto di lavoro (cambiando efficienza e purezza) lungo la curva denominata "Receiver Operation Characteristics"
- Come scegliere il taglio ? Occorre conoscere le probabilità a priori (Priors), cioè le abbondanze di Segnale e Background

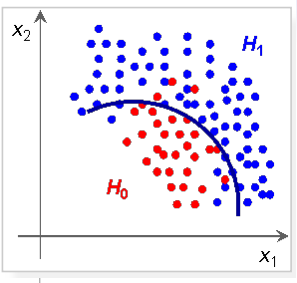
La Classificazione di Eventi nella Realtà

- Sfortunatamente non sempre si conoscono le densità di probabilità del Segnale e del Background
- Spesso si utilizza la generazione Monte Carlo per avere una stima del Segnale e del Fondo
- Altre volte si riescono a trovare campioni di eventi di dati reali che rappresentano il segnale e il fondo
- Utilizziamo gli eventi di "training" per
  - Dare una stima della forma funzionale per $P(x|C)$ dalla quale si può ricavare il rapporto di verosimiglianza
  - Determinare una funzione di discriminazione $y(x)$ ed una regione dove tagliare per ottimizzare la separazione del Segnale dal Background
- anche se non esiste una prescrizione magica che vada bene in generale
- $\Rightarrow$ supervised (machine) learning

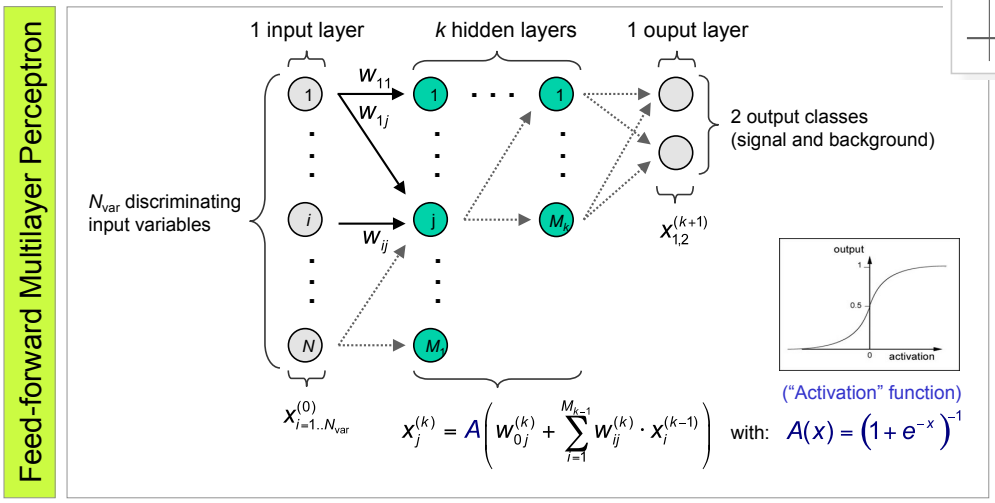
Metodi di Analisi Multivariata

- Alcune tecniche di analisi
  - Rectangular Cut optimization (binary splits)
  - Projective and multidimensional likelihood estimation
  - Linear and nonlinear discriminant analysis ((H-Matrix, Fisher, FDA)
  - Artificial Neural Networks
  - Boosted/Bagged decision trees
  - Predictive learning via rule ensambles (RuleFit)
  - Support Vector Machine
- TMVA è un ambiente di lavoro per Analisi Multivariata e Machine Learning (diponibile in ROOT) che fornisce
  - Una interfaccia comune per svariati metodi di classificazione e regressione
  - Home Page: http://tmva.sourceforge.net
  - Tutorial Wiki: https://twiki.cern.ch/twiki/bin/view/TMVA/WebHome
  - User Guide: http://tmva.sourceforge.net/docu/TMVAUsersGuide.pdf

Artificial Neural Networks



- tecnica Multivariata nonlineare: permette di ottenere una classificazione "attivando" specifici nodi di output tramite pesi nonineari



Reti Neurali disponibili in TMVA:

- TMlpANN: Interface to ROOT MLP implementation
- MLP: TMVA own MLP implementation for increased speed and flexibility
- CFMlpANN: ALEPH Higgs search ANN, translated from FORTRAN

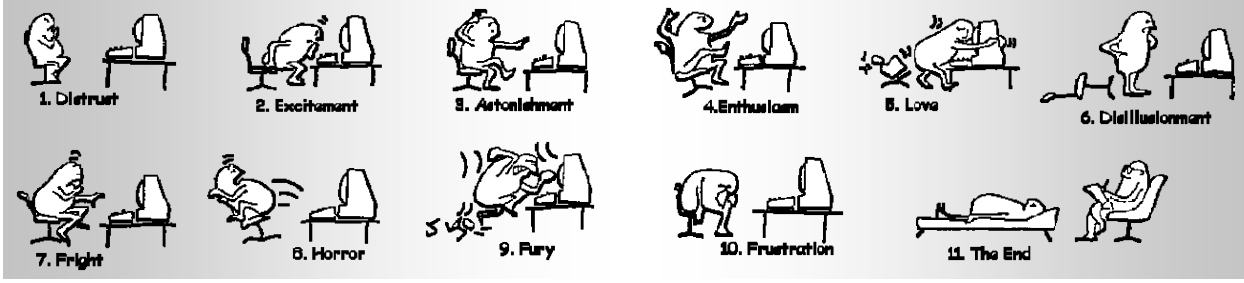
Using TMVA

Helge Voss

A typical TMVA analysis consists of two main steps:

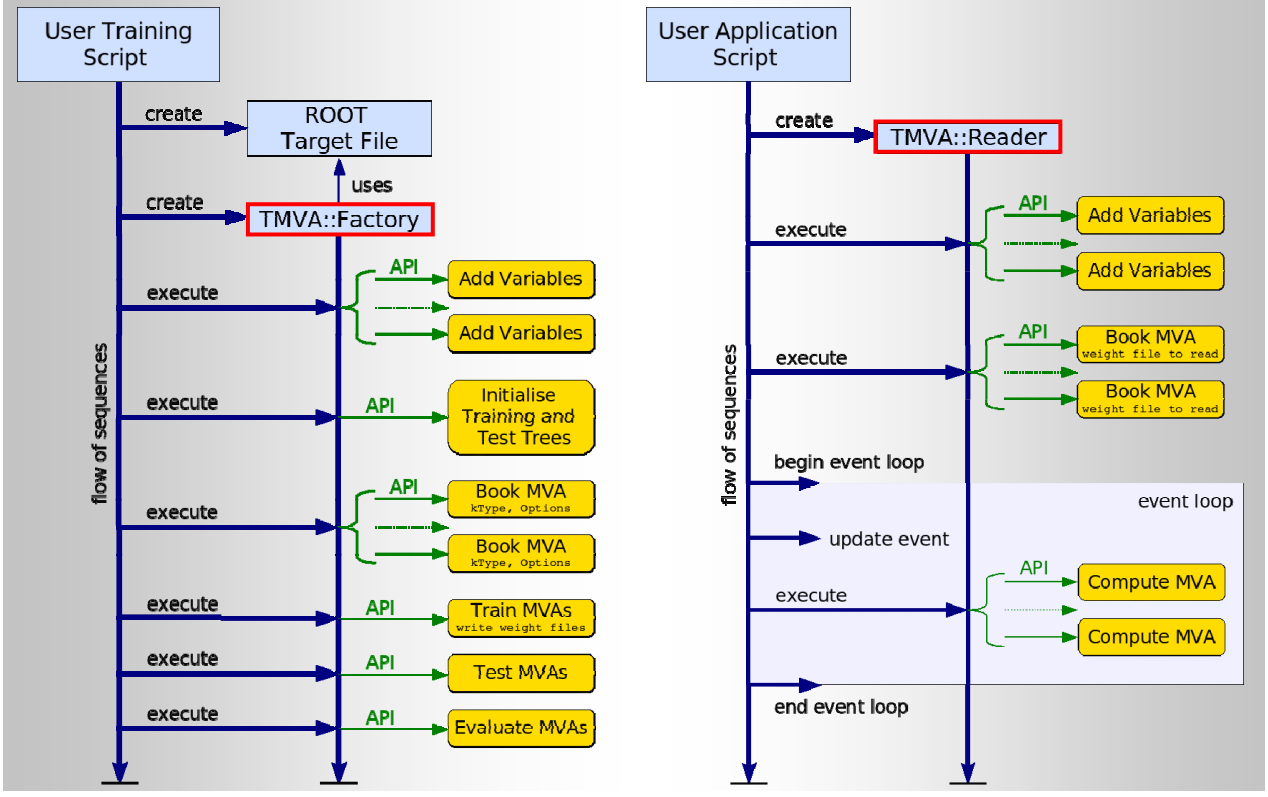
1. *Training phase*: training, testing and evaluation of classifiers using data samples with known signal and background composition
2. *Application phase*: using selected trained classifiers to classify unknown data samples

- Illustration of these steps with toy data samples



Code Flow for *Training* and *Application* Phases

Helge Voss



A Simple Example for *Training*

Helge Voss

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

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

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

   factory->AddSignalTree         ( (TTree*)input->Get("TreeS"), 1.0 );
   factory->AddBackgroundTree ( (TTree*)input->Get("TreeB"), 1.0 );

   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

← register input variables

← select MVA methods

← train, test and evaluate

→ TMVA tutorial

A Simple Example for an *Application*

Helge Voss

```
void TMVClassificationApplication( )
{
   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 classifier",  "weights/MVAnalysis_MLP.weights.txt" );

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

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

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

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

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

← create *Reader*

← register the variables

← book classifier(s)

← prepare event loop

← compute input variables

← calculate classifier output

→ TMVA tutorial

Data Preparation

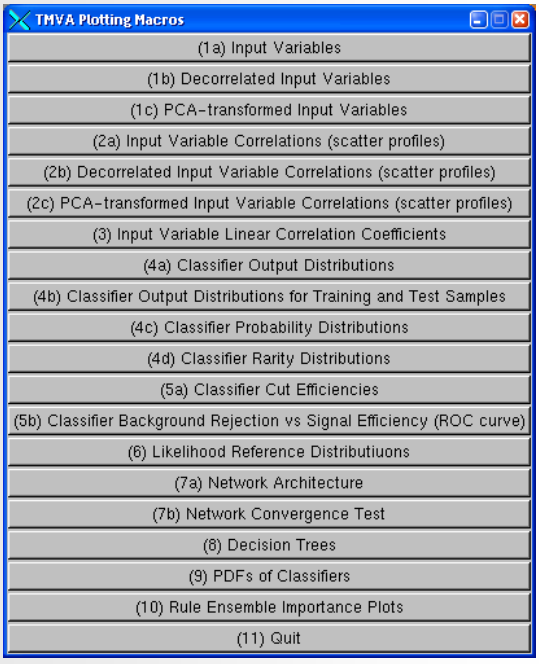
Helge Voss

- Data input format: ROOT TTree or ASCII
- Selection any subset or combination or function of available variables
- Apply pre-selection cuts (possibly independent for signal and bkg)
- Define global event weights for signal or background input files
- Define individual event weight (use of any input variable present in training data)
- Choose one out of various methods for splitting into training and test samples:
  - Block wise
  - Randomly
  - Periodically (*i.e.* periodically 3 testing ev., 2 training ev., 3 testing ev, 2 training ev. ….)
  - User defined training and test trees
- Choose preprocessing of input variables (*e.g.*, decorrelation)

MVA Evaluation Framework

Helge Voss

- TMVA is not only a collection of classifiers, but an MVA framework
- After training, TMVA provides ROOT evaluation 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

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