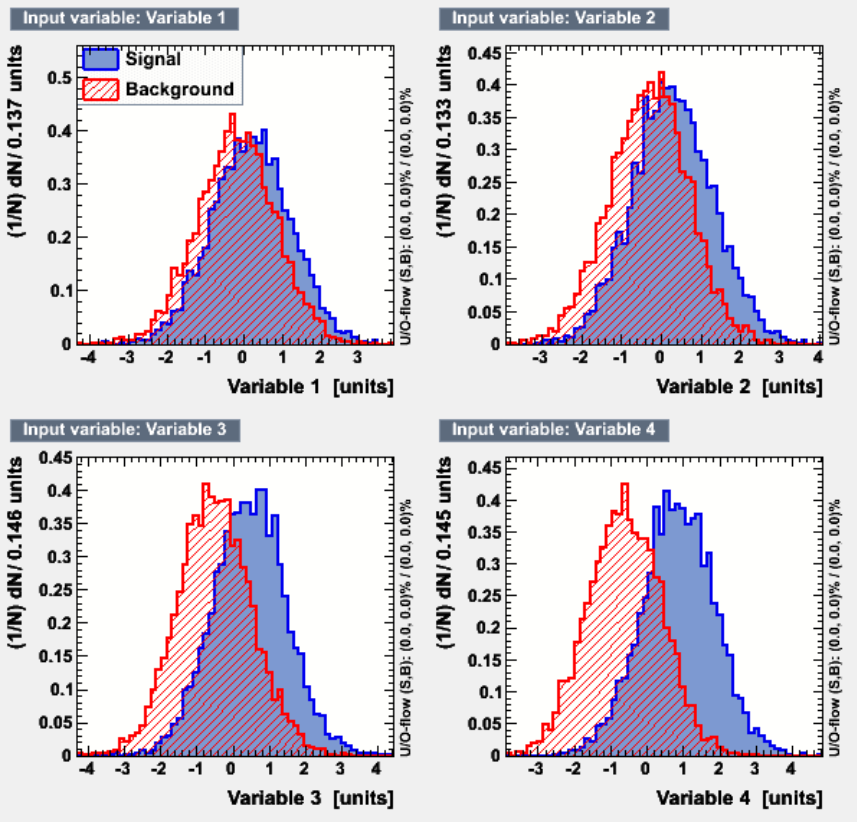
Analysis Centre Statistics School 2014 Hamburg
Multivariate Analysis Tutorial session – Part A

1 Event Classification by eye

Consider two events, described by four variables with the following values:

$$e_1 = (0.0, -1, 2.5, 2)^T \quad (1)$$
$$e_2 = (0.2, 0.2, 0.3, 1)^T \quad (2)$$

Given the signal and background distribution below, classify the events as signal-like or background-like.



1. Which event is more likely to be signal?

2. Consider a sequence of cuts to classify events as signal. Which one of the two events is more likely to fail this cut-based analysis?

3. Can you think of alternative techniques to discriminate events based on the given information, other than a cut-based approach?

General remarks on the computer exercise

There are four tutorials which focus on the MVA methods. Tutorials –A–, –B–, –C– and optional exercise –E– concentrate on the main application of multivariate techniques: classification. You will be using ROOT, and in most parts TMVA (Toolkit for Multivariate Data Analysis). Tutorial –D– focuses on another central application of MVA techniques - function estimation, or regression.

During the statistics school, log into one of the NAF login servers. Initialize the ROOT environment by invoking

```
$ module load root
```

In order to suppress (harmless) error messages, before using the TMVA scripts invoke at the start of your work session (but after the module was loaded)

```
$  export TMVASYS=$ROOTSYS
```

Please copy and un-tar (`tar xf tmvaExercises.tar`) the exercises from the common afs space (they also will be available from the agenda page) at

```
/afs/desy.de/group/school/statschool2014/tmvaExercises.tar
```

This will create a directory

```
~/tmvaExercises/
```

You will need to modify some ROOT macros in order to perform the tasks. To have the initial, untouched version also available, there is a sub directory, in which the unmodified versions are present:

```
~/tmvaExercises/unmodifiedMacros/
```

All basic macros contain hints, where lines need to be edited.
As a side note, the TMVA scripts are taken from the current ROOT release (5.34.18), from the `$ROOTSYS/tmva/test` directory.In non-tutorial situation, one would probably use a compiled version; but the interactiveness would be severely hampered.

2 TMVA exercise – Classification

The intention of this exercise is to familiarize you with the concept of multivariate input distributions. The input data set is based on Gaussian distributions, with added correlations. The data file is `data_correlatedGauss.root`, and the macros are `FindCuts_correlatedGauss.C` and `TMVAClassification_correlatedGauss.C`.

2.1 Modifying cuts manually

The macro `FindCuts_correlatedGauss.C` can be used to

- plot the input distributions as they are in the data file,
- impose cuts on the variables (by modifying the source code),
- plot the variable distributions after the cuts have been applied and
- retrieve the efficiency and background rejection connected to the cuts (will be printed on the command line).

Run the macro with the usual commands, e.g. from the command line
`$ root FindCuts_correlatedGauss.C`
or if you are already in an interactive root session by
`root[0] .x FindCuts_correlatedGauss.C`
The tasks:

1. Play around with the macro and try to find a set of cuts on the four variables that has a good value for efficiency and purity.
2. Try to find the best (highest) value for signal efficiency for the given background rejection of 80%.

2.2 Use Genetic Algorithm to determine cuts

In TMVA several cut methods are available. One of them uses a Genetics Algorithm for maximizing the background rejection for a given signal efficiency. You can use one of the prepared macros to invoke this algorithm.

1. Invoke
   `$ root TMVAClassification_correlatedGauss.C\(\"CutsGA\"\)`
   and look at the result (the GUI will pop up)
2. Compare your result in terms of performance with the one determined by the GA algorithm in TMVAClassification.
3. Take a look at the weights file
   `weights/TMVAClassification_correlatedGauss_CutsGA.weights.xml`
   Can you find the *best* efficiency value? Why is this value the best?