An example of event analysis: a likelihood discriminator

P. Ronchese
Dipartimento di Fisica e Astronomia “G.Galilei”

Università di Padova

“Object oriented programming and C++” course

Events

The files `train.txt` and `test.txt` contain the data, corresponding to the output of a detector for particles that can be muons or background (mostly pions).
Muons are particles similar to electrons with a mass $\sim$ 200 times larger.

For each event the input files contain:

- an identifier
- 7 numbers corresponding to different quantities
- a flag (1 or 0) for muons or background

the input files are to be used as follows:

- `train.txt` is to be used for training, i.e. to let the system learning how to discriminate muons and background
- `test.txt` is to be used to test the system

Event discrimination - version 1

Build a likelihood discriminator to classify events as “signal” and “background”.

- Build a set of discriminating variables starting from the quantities read from the input file(s).
- Create and fill histograms of all the variables for signal and background events.
- For each variable get from the histograms the probabilities $P_{j,\mathrm{sig}}$ and $P_{j,\mathrm{bkg}}$ that the variable stays inside a defined interval, for a signal or background event.
- Compute the discriminating variable:

$$D = \Pi_j P_j = \frac{\Pi_j P_{j,\mathrm{sig}}}{\Pi_j P_{j,\mathrm{sig}} + \Pi_j P_{j,\mathrm{bkg}}} \ .$$

- Create and fill histograms of the discriminating variable for signal and background events.

Event discrimination - version 2

Build a likelihood discriminator to
classify events as “signal” and “background”.

- Modify the version 1 to encapsulate the following operations for a generic set of variables:
    - compute the discriminating variable,
    - fill and save on file the variables histograms,
    - read from file the variables histograms.
- Modify the concrete discriminator `class` to use the generic one by inheritance.

Event discrimination - version 3

Test the likelihood discriminator
with different subsets of variables.

- Modify version 2 to compare the discriminating performances with different subsets of variables:
- Create several discriminator `class`es, setting different variables as “active”.
- Create a corresponding histogram for each discriminator.
- For each event compute the discriminator for all the variable choices and fill the corresponding histograms.

Event discrimination - version 4

Test the likelihood discriminator
with different subsets of variables.

- Modify version 3 to compute the variables only once for all the discriminators.
- Move the variable declaration and computation to a new small `class` .
- Include in the discriminator `class` a pointer to this new `class` .
- Provide in the constructor of the discriminator `class` the choice to create a new set of variables or share the ones in another object.