Multivariate analysis

Alexander Khanov

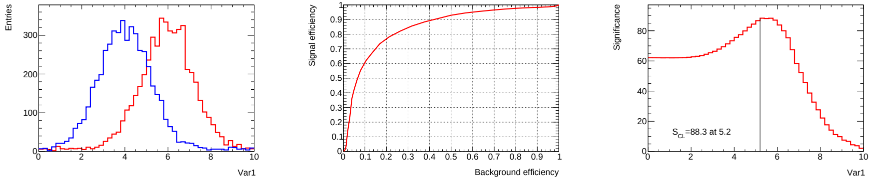
PHYS6260: Experimental Methods is HEP
Oklahoma State University

November 1, 2023

Separation of signal from background

- 1d case: straightforward
  - plot signal and background distributions of the discriminating variable
  - optimize the cut to obtain the best sensitivity
- Approximate figure of merit: significance $S = s/\sqrt{b}$
- A better figure of merit: optimizing the likelihood ratio L(S+B)/L(B)

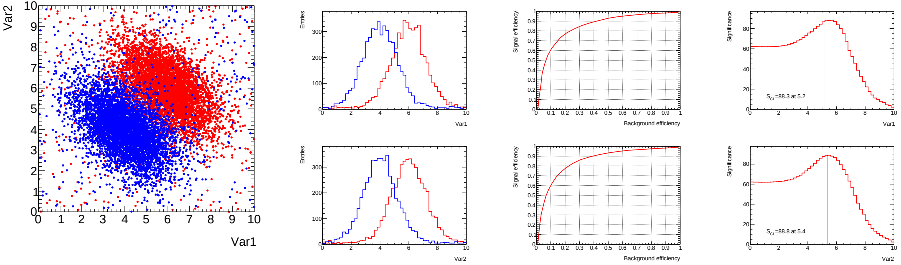
$$S_{\mathrm{CL}} = \sqrt{2((s+b)\ln(1+s/b) - s)}$$



- What to do if there are more than one input variable?

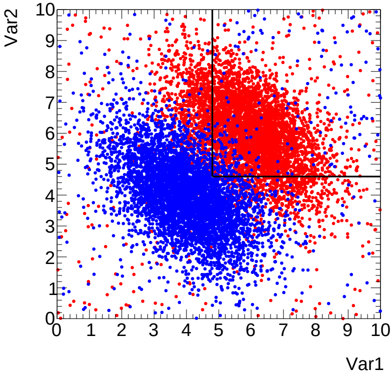
Multivariate case

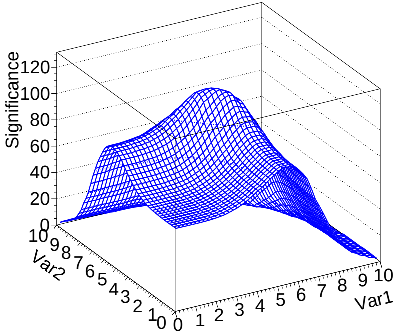
- There are typically $\gg 1$ variables
  - it's not easy to see the overall picture
- Some of the variables may be correlated



Grid search (a.k.a. cut and count)

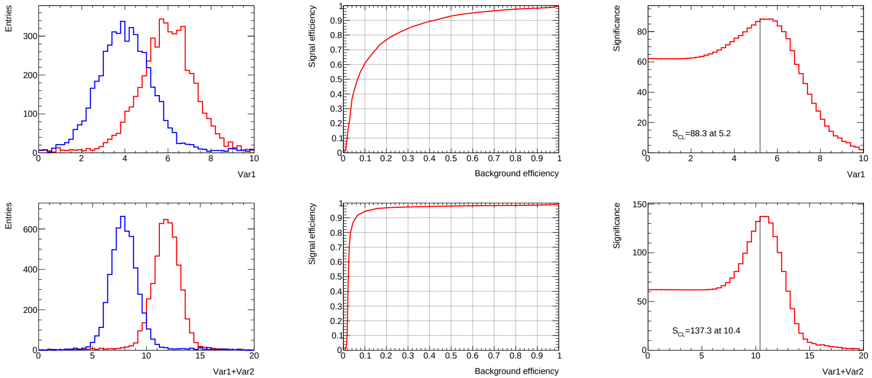
- Try all combinations of cuts, pick the one that provides the best significance





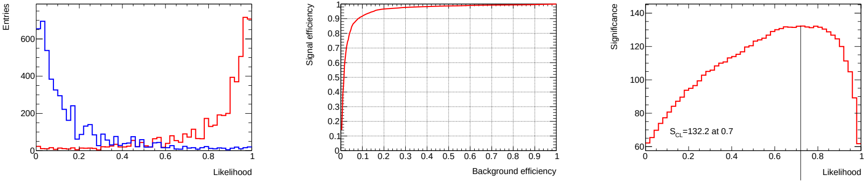
Linear discriminants

- Fisher discriminant: $F = \sum w_i x_i$
  - weights w_i are chosen in such a way that to optimize the separation
- In our example, $F = \mathrm{Var1} + \mathrm{Var2}$ works the best



Likelihood

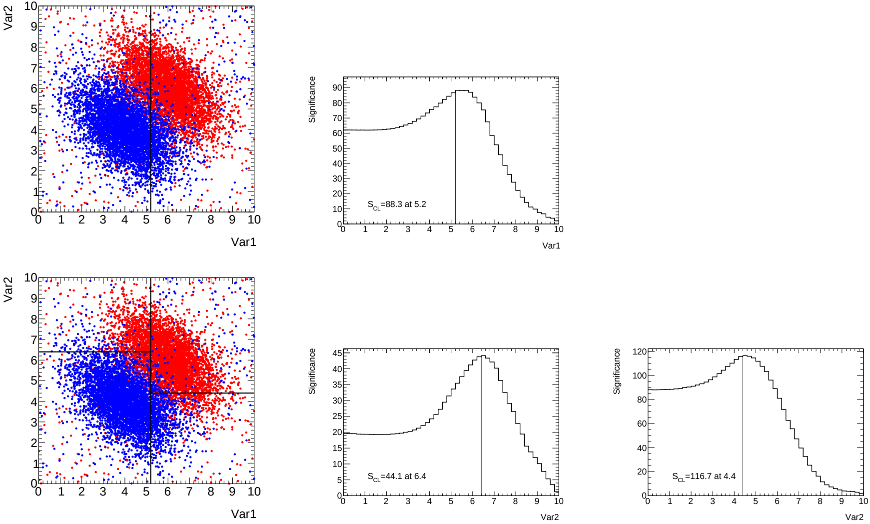
- Given pdf's for signal $s = \prod s_i$ and background $b = \prod b_i$, the likelihood discriminator is $L = \dfrac{s}{s+b}$



- Simple likelihood doesn't work well if the variables are correlated
  - a variation of the method transforms the variables such that their correlation matrix becomes diagonal
  - this is a linear approximation, so not perfect

Decision trees

- Optimize one cut at a time, split the sample into subsets

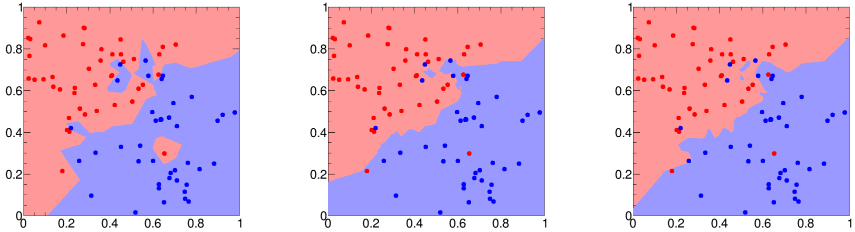


Boosted decision trees

- The idea is to combine many weak learners (trees trained on random subsets of the training sample) into a powerful classifier
- The simplest approach is to train an ensemble of trees (all in parallel) and combine their inputs by the majority vote
  - this approach is known as random forests
- Boosting is a method based on iterative tree training
  - the output of the algorithm is a weighted sum of all trees trained so far
  - each event used for training is also assigned a weight based on how "difficult" it is: the events that are misclassified get their weight increased and vice versa

k nearest neighbors

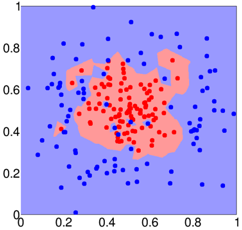
- This is an example of a nonparametric method
  - the effective number of model parameters grows with the data set size
- Training sample: a set of labeled points
- The algorithm: for each point $\mathbf{x}$ to be classified, its label is defined as majority of labels among k points from the training sample that are closest to $\mathbf{x}$



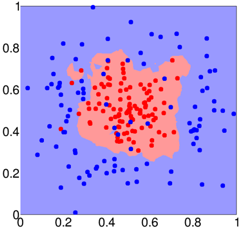
$k = 1$ $\quad$ $k = 3$ $\quad$ $k = 5$

k nearest neighbors (2)

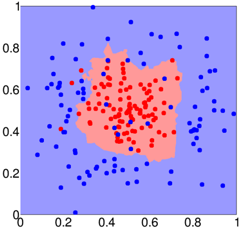
- The method efficiently works with complicated topologies
- It can be used for regression: the value assigned to the points is calculated as mean of its k closest neighbor values



$k = 1$



$k = 3$



$k = 5$