Multi-dimensional likelihood

Bill Murray
CCLRC RAL
w.murray@rl.ac.uk
home.cern.ch/~murray

ATLAS UK Higgs meeting
RAL
17 June, 2006

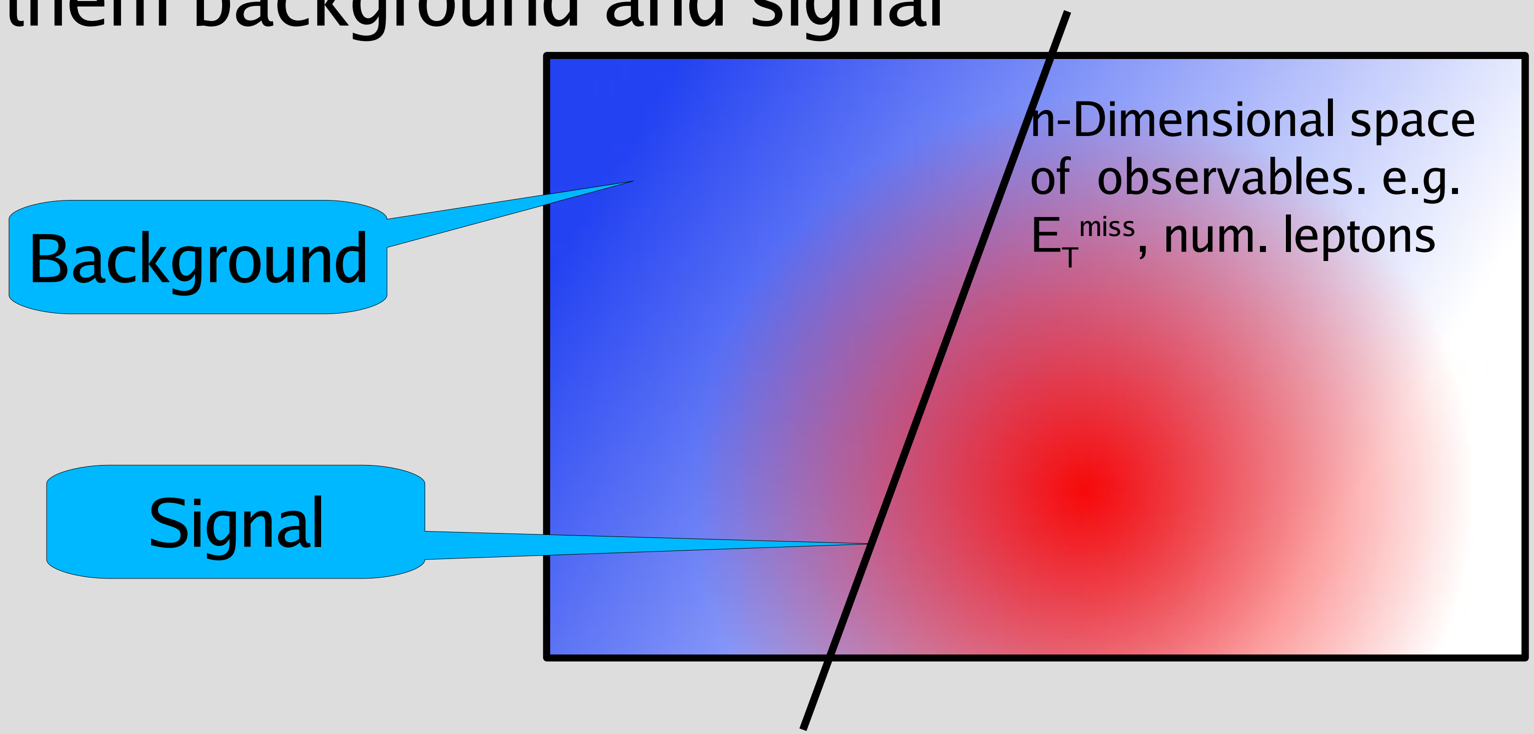
Likelihood
An n-D implementation
Application to b-tag

What is likelihood?

- Likelihood is the product, over all observed events, of the probability of observing those events.
- Maximising the likelihood by varying model parameters is extremely powerful fit technique
- Separation of events into signal and background according to ratio of likelihood of the hypotheses is optimal technique

The right answer

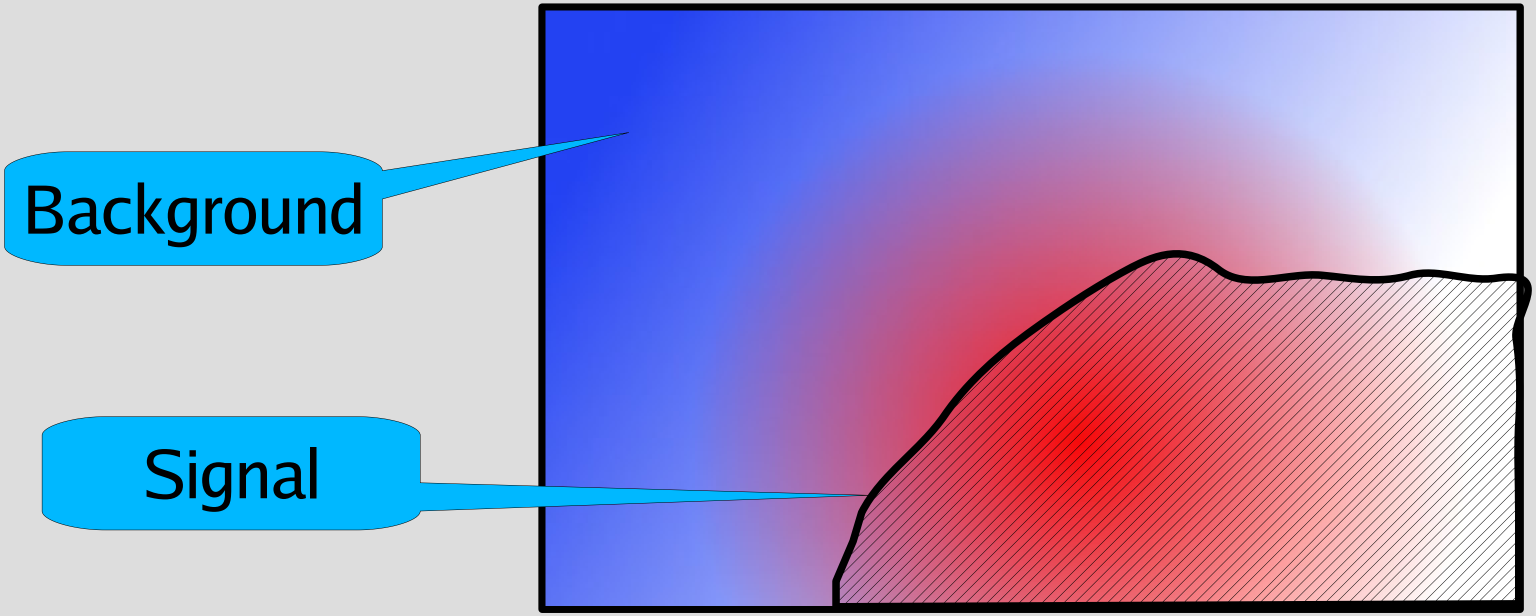
- Consider separating a dataset into 2 classes
  - Call them background and signal



- A simple cut is not optimal

The right answer II

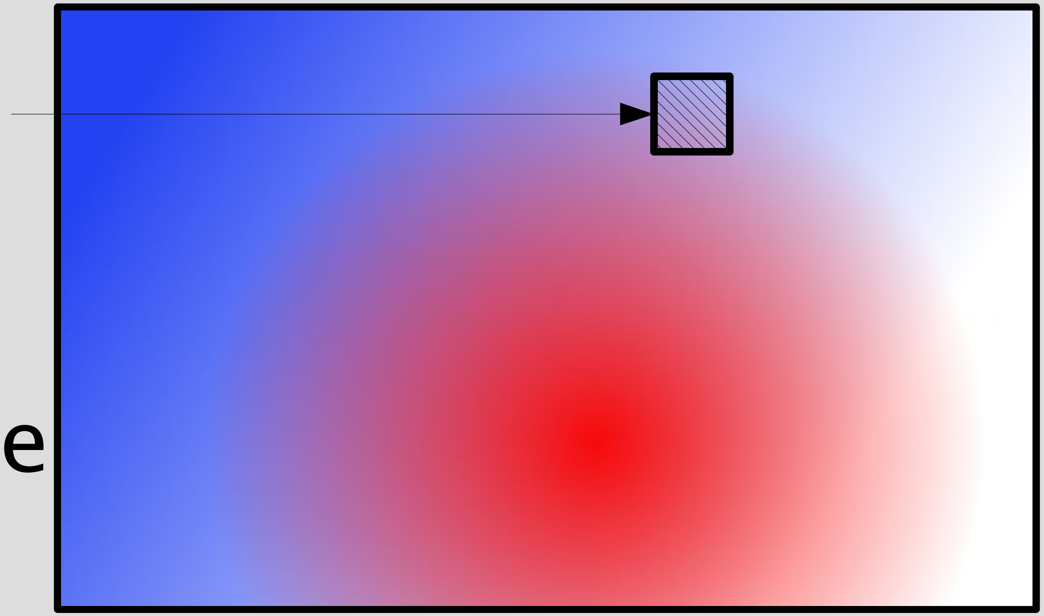
- What is optimal?



- Maybe something like this might be...

The right answer III

- For an given efficiency, we want to minimize background
- Sort space by s/b ratio in small box
- Accept all areas with s/b above some threshold



- This is the ***Likelihood Ratio***

The right answer

- For optimal separation, order events by

$$L_s/L_b$$

  - Identical to ordering by

$$L_{s+b}/L_b$$

- More powerful is to fit in the 1-D space defined above.

This is the right answer

Determination of s, b densities

- We may know matrix elements
  - Not for e.g. a b-tag
  - But anyway there are detector effects
- Usually taken from simulation

Using MC to calculate density

- Brute force:
  - Divide our n-D space into hypercubes with m divisions of each axis
  - m^n elements, need 100 m^n events for 10% estimate.
  - e.g. 1,000,000,000 for 7 dimensions and 10 bins in each
- This assumed a uniform density – actually need far more
  - The purpose was to separate different distributions

Better likelihood estimation

- Clever binning
  - Starts to lead to tree techniques
- Kernel density estimators
  - Size of kernel grows with dimensions
  - Edges are an issue
- Ignore correlations in variables
  - Very commonly done ‘**I used likelihood**’
- Pretend measured=true, correct later
  - Linked to OO techniques, bias correction
  - Problem for b-tag: There is no true

Alternative approaches

- Cloud of cuts
  - Use a random cut generator to optimise cuts
- Neural nets
  - Well known, good for high-dimensions
- Support vector machines
  - Computationally easier than kernel
- Decision trees
  - Boosted or not?

Kernel Likelihoods

- Directly estimate PDF of distributions based upon training sample events.
- Some kernel, usually Gaussian smears the sample – increases widths
- Fully optimal iff infinite MC statistics
- Metric of kernel, (size, aspect ratio) hard to optimize
  - Watch kernel size dependence on stats.
- Kernel size must grow with dimensions;
  - Lose precision if unnecessary ones added
- Big storage/computational requirements

Conclusions

- The likelihood ratio underpins everything
  - Consider using it
  - Neural nets are a good tool too of course
  - Boosted trees very interesting?
- Cost of computing becoming important
  - Optimal methods not *necessarily* optimal
  - But the data is very expensive too.
- A general purpose density estimator would be very useful
  - See next talk!