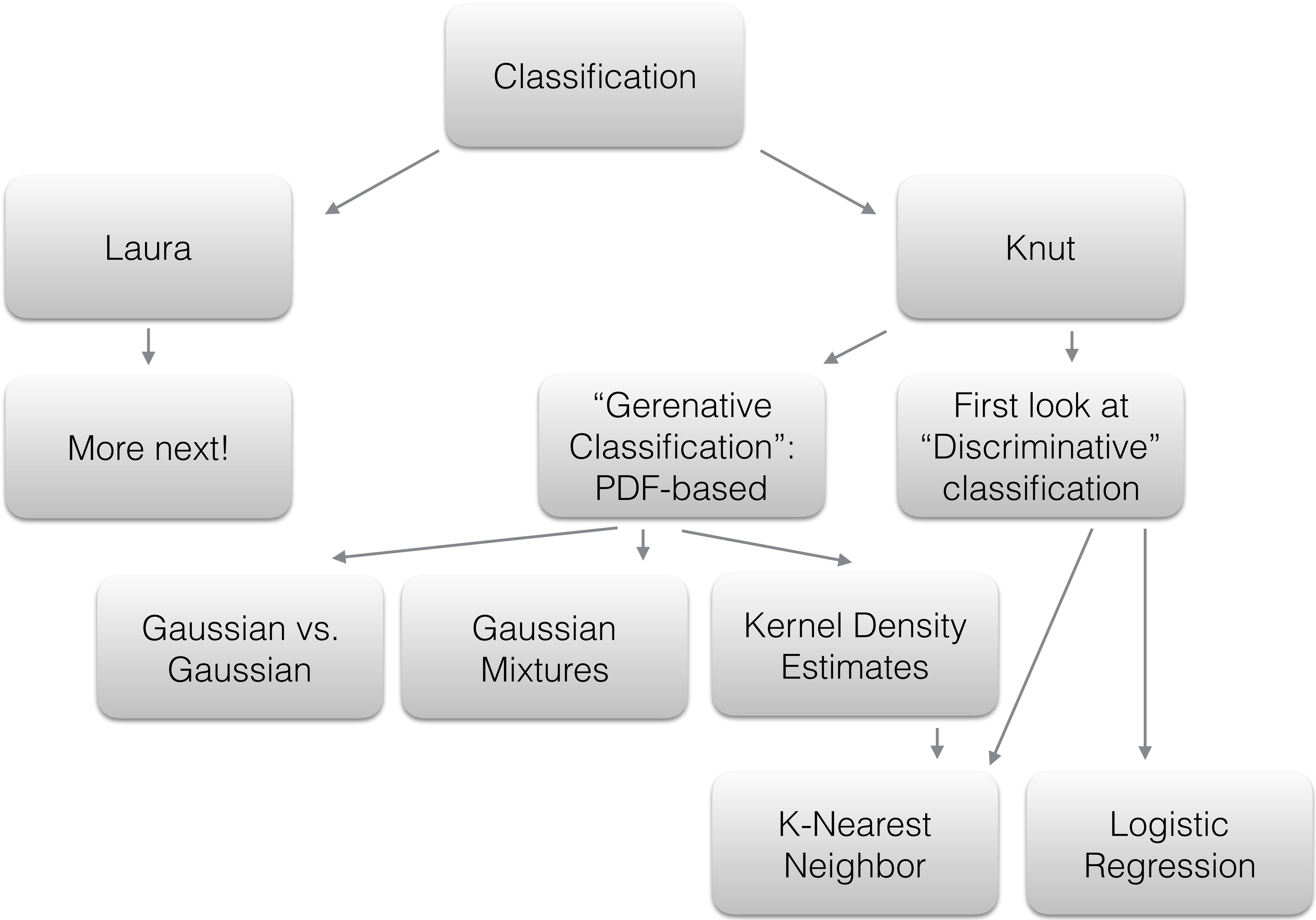
classification, pt 1

knut.mora@fysik.su.se

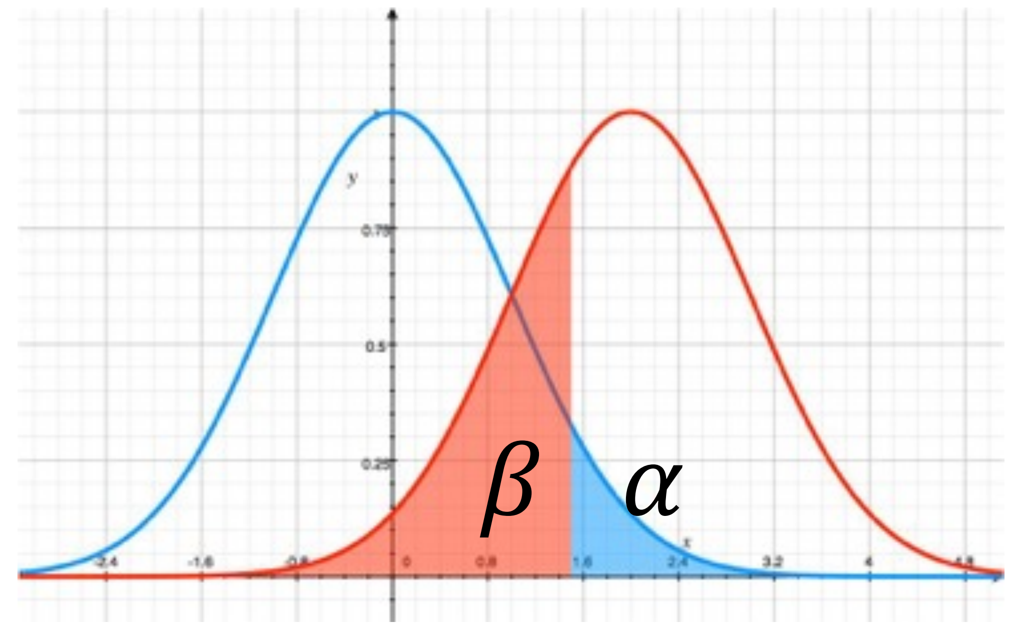

Classification
Laura
Knut
More next!
“Gerenative Classification”: PDF-based
First look at “Discriminative” classification
Gaussian vs. Gaussian
Gaussian Mixtures
Kernel Density Estimates
K-Nearest Neighbor
Logistic Regression

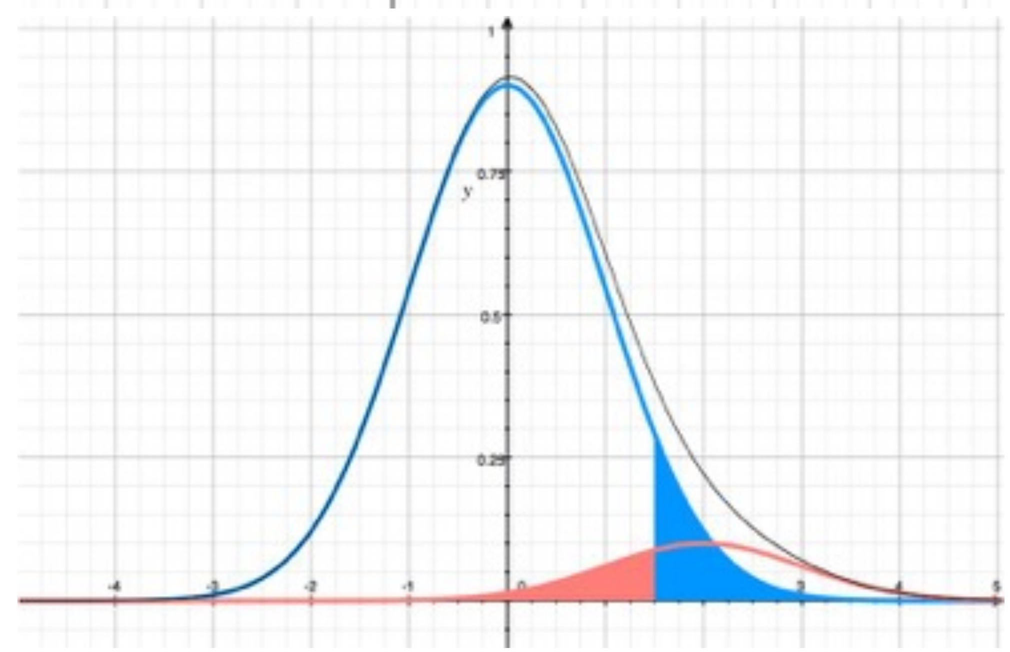
setup/terminology

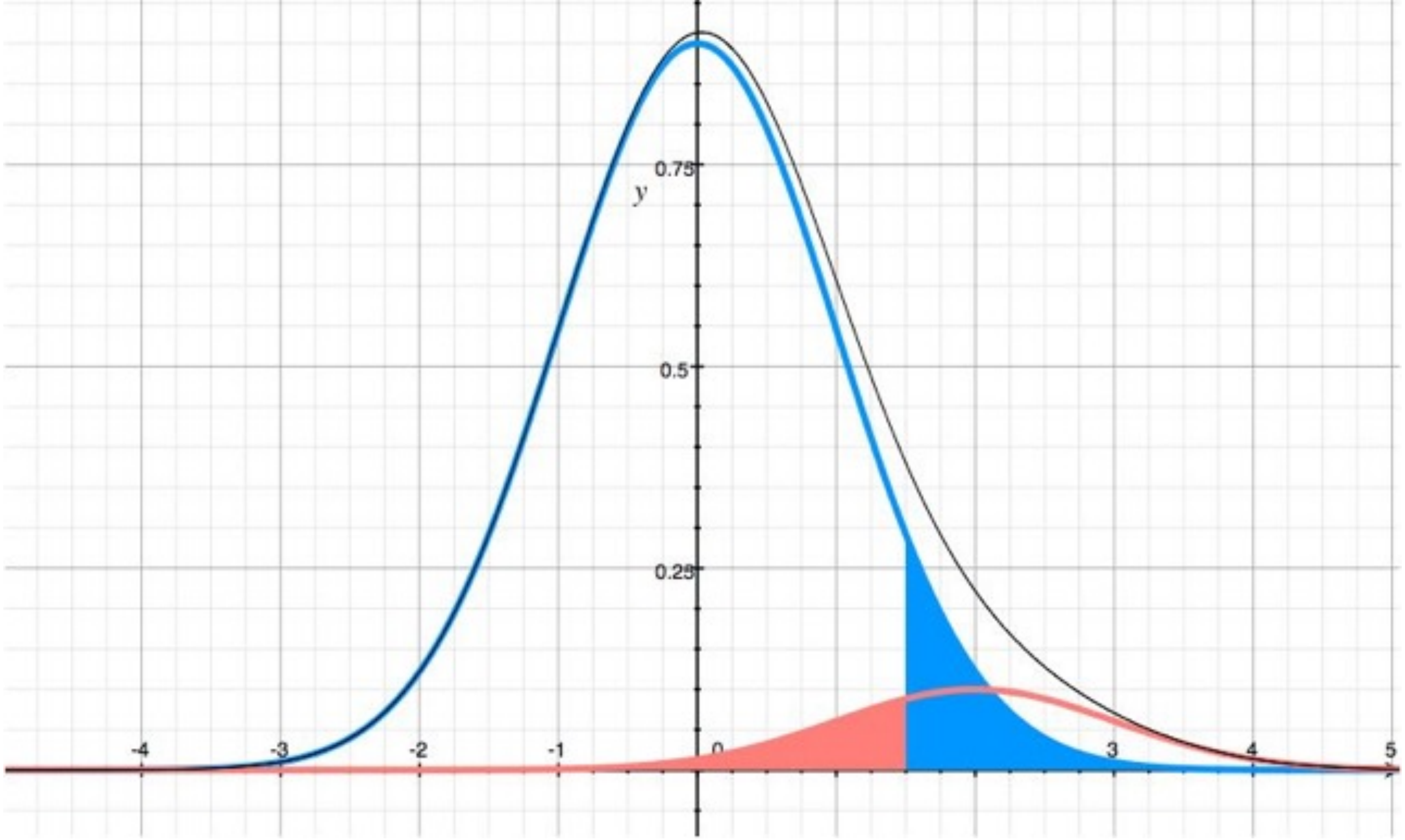
- In hypothesis testing, α and β is the probability of false negatives and positives given H0 and H1, respectively.

- For classification, the probability of each class influences the probability of misclassification.

- Also note that like in in hypothesis testing, α is a fiducial parameter you choose







Two measures of performance:

- Completeness, the probability of classifying signal (A) correctly; pA*(1-α)
- Contamination, the probability that an event classed as signal (A) is background (B);

	class A	class B
Probability:	pA	pB
classed as A	pA*(1-α)	pB*β
classed as B	pA*α	pB*(1-β)

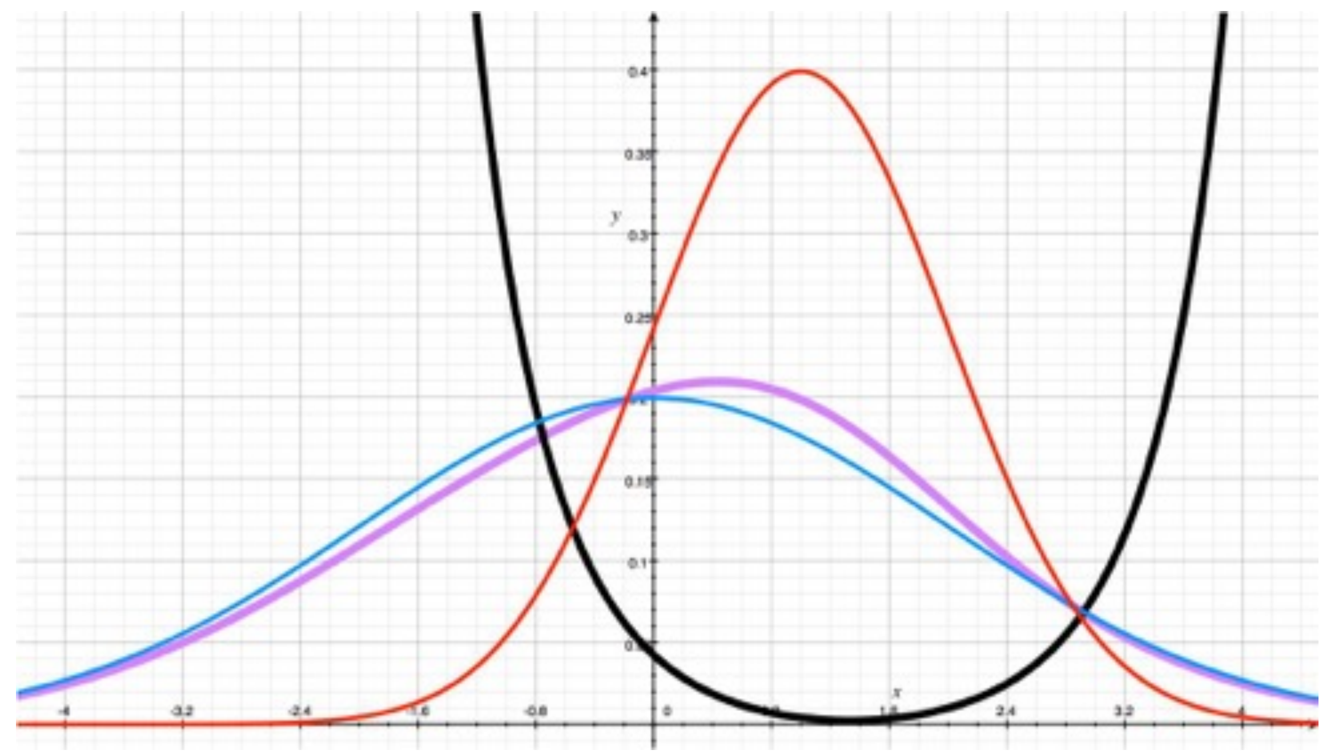
likelihood ratios

- _If_ you know the PDF, there is a guaranteed best solution (courtesy of the Neyman-Pearson lemma): cut on ll=L(h0|x)/L(h1|x)

- Background (blue): mu=0,sigma=2

- Background (red): mu=1,sigma=1

- The black line shows the likelihood ratio



$$ll = \log\left[\frac{\frac{1}{\sqrt{2\pi}\sigma_1} e^{\frac{(x-\mu_1)^2}{-2\cdot\sigma_1^2}}}{\frac{1}{\sqrt{2\pi}\sigma_0} e^{\frac{(x-\mu_0)^2}{-2\cdot\sigma_0^2}}}\right]$$

classification using pdfs

- You can also use Bayes thm. to invert the pdfs given each class to give a probability for an individual event to be each class

$$p(y_k|\vec{x}) = \frac{p(\vec{x}|y_k)p_k}{\sum p(\vec{x}|y_k)p_k}$$

- As with regression, one may construct an estimator for y (0=signal, 1=background, say), g:

$$g(\vec{x}) = \hat{y} = \int y \cdot p(y|\vec{x})dy$$

- The book continues to cut on g=0.5 (makes sense _if_ the two classes are equally important to you)

naive bayes

- Assumptions makes life easier
- Naive Bayes: assume that each variable in x is uncorrelated.
- Afterwards, choose the y that maximises p(y|x)
- estimating p(x|y) may be fitted, estimated etc.

gaussian bayes

- (Naive and otherwise)
- Fit an (uncorrelated) multivariate gaussian to x for each training sample
- find the class has the largest probability given the event data
- Correlations are computationally expensive

linear discriminant

- Assume: in each variable, class distributions are gaussians with equal covariance and different means
- Only a linear term in g if this is the case
- I think there must be a misprint in 9.25
- If the covariance is allowed to vary- a quadratic boundary is achieved

kernel density estimation

- Leaving gaussians, estimating the pdf directly using kernels allows flexibility to represent complex shapes, and must be optimized only in the fiducial bandwidth

- Note that the optimization should (can) be done directly with respect to the discrimination power

- Clever branching in the KDE computation is possible.

k-nearest neighbor

- aka conformism:
- If the nearest training point is of class y, so is the event!
- smoother: average class of k nearest points

directly estimating decision boundaries

- Like the nearest-neighbor solution , these do not depend on finding a pdf estimate

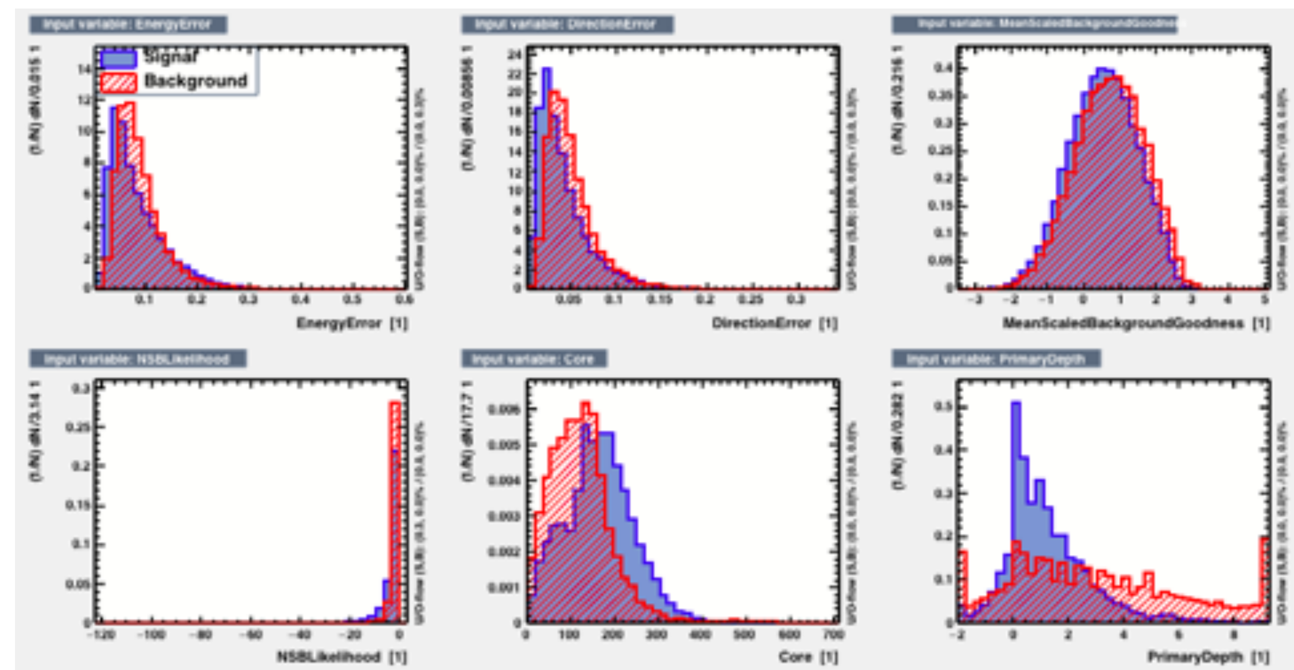
- Another example is logistic regression, where a weighted sum of xi is put into a sigmoid curve

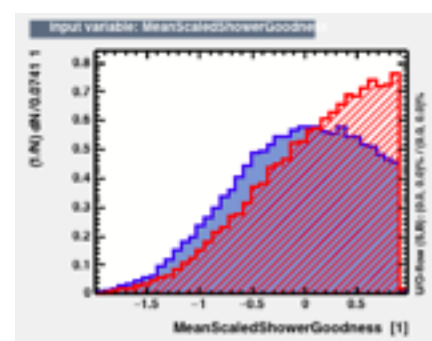
- Theta is minimized with respect to classification error

$$p(y|x) = \frac{[e^{\sum \theta_i x_i}]}{1 + e^{[\sum \theta_i x_i]}}$$

exercise/fun

- On the github, there are two files, labeled ClassSample_training_N.npy
- N=0 is background taken from data, N=1 is photon MC
- ClassificationReadme.txt contains metadata
- To the right is some plots that root TMVA spits out when classifying these.

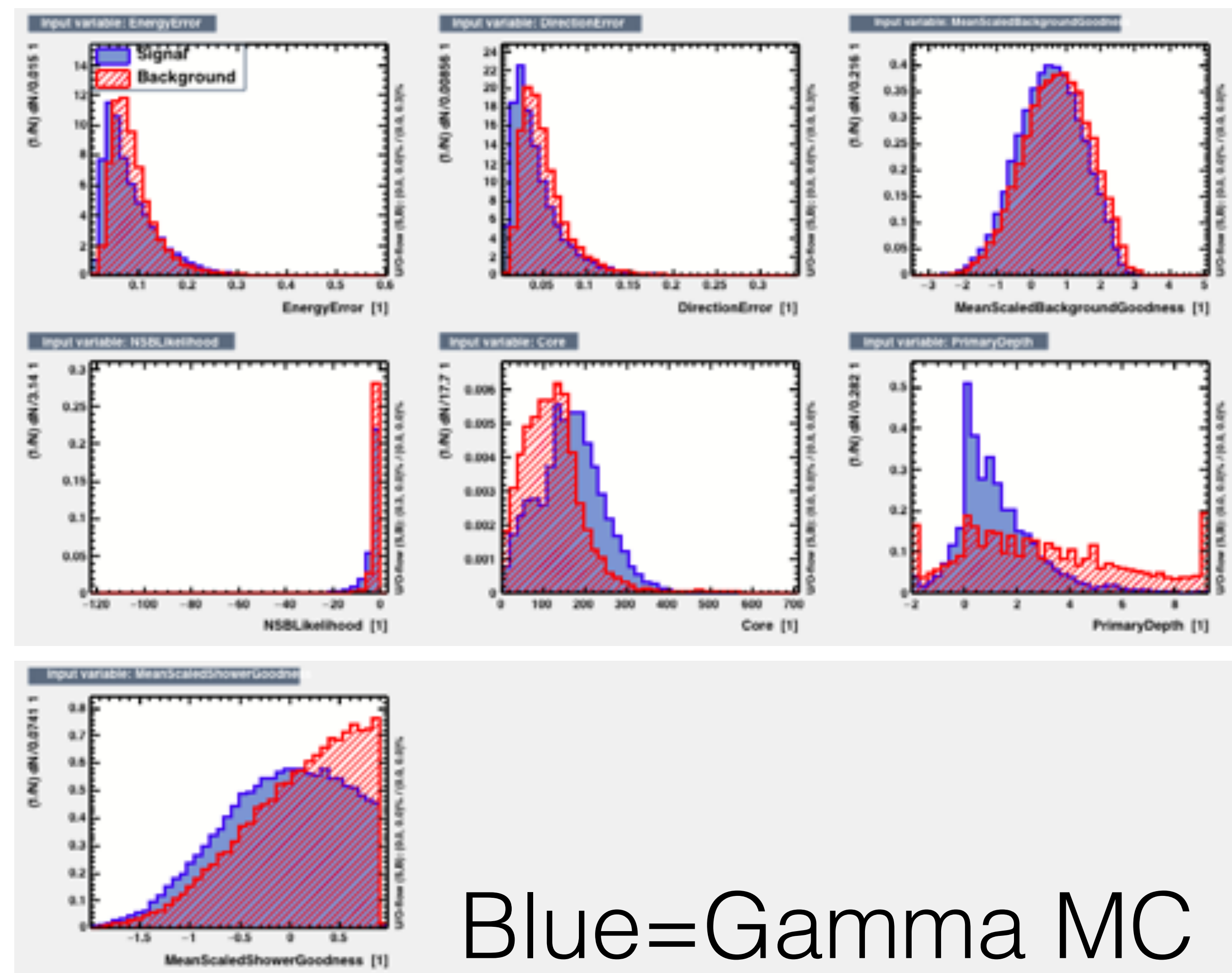




Blue=Gamma MC
Red=OFF data

exercise/fun

- Proposal exercise:
  - One group to perform Gaussian bayes
  - One group to use one of the PDF-estimation tools from before (say, KDE if computationally feasible, gaussian mixtures otherwise)
  - One to use Logistic Regression
- All group to compute a set of N (between 5 and 1000) points of true positive pA*(1-alpha) vs false positive pB*(beta) to be plotted and compared.



Blue=Gamma MC
Red=OFF data