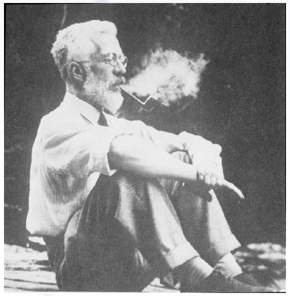
Machine Learning

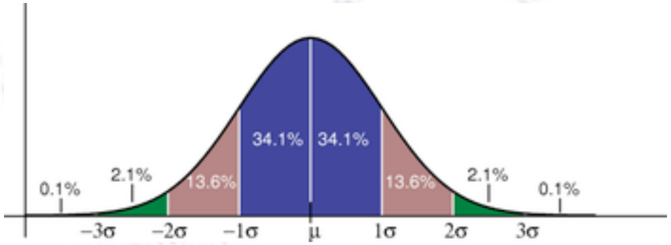
Introduction to MultiVariate Analysis








Troels C. Petersen (NBI)



"Statistics is merely a quantisation of common sense - Machine Learning is a sharpening of it!"

Dimensionality and Complexity

Humans are good at seeing/understanding data in few dimensions!
However, as dimensionality grows, complexity grows exponentially ("curse of dimensionality"), and humans are generally not geared for such challenges.

	Low dim.	High dim.
Linear	Humans: Computers:	Humans: Computers:
Non-linear	Humans: Computers:	Humans: Computers:

Computers, on the other hand, are OK with high dimensionality, albeit the growth of the challenge, but have a harder time facing non-linear issues.

However, through smart algorithms, computers have learned to deal with it all!
That is essentially what Machine Learning has enabled!

Dimensionality and Complexity

Humans are good at seeing/understanding data in few dimensions!
However, as dimensionality grows, complexity grows exponentially ("curse of dimensionality"), and humans are generally not geared for such challenges.

	Low dim.	High dim.
Linear	Humans: ✓ Computers: ✓	Humans: ÷ Computers: ✓
Non-linear	Humans: Computers:	Humans: Computers:

Computers, on the other hand, are OK with high dimensionality, albeit the growth of the challenge, but have a harder time facing non-linear issues.

However, through smart algorithms, computers have learned to deal with it all!
That is essentially what Machine Learning has enabled!

Dimensionality and Complexity

Humans are good at seeing/understanding data in few dimensions!
However, as dimensionality grows, complexity grows exponentially ("curse of dimensionality"), and humans are generally not geared for such challenges.

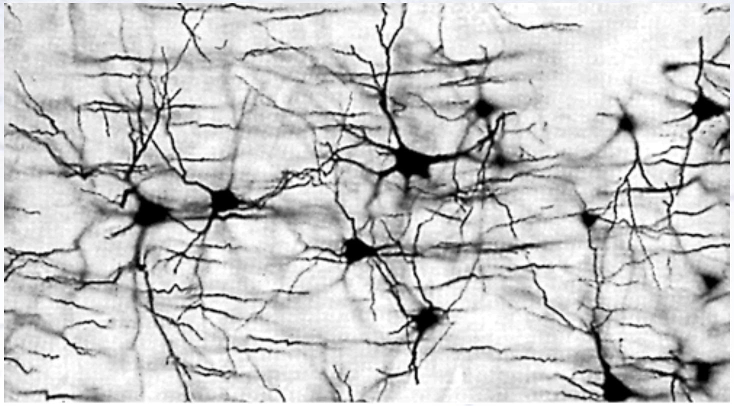
	Low dim.	High dim.
Linear	Humans: ✓ Computers: ✓	Humans: ÷ Computers: ✓
Non-linear	Humans: ✓ Computers: (✓)	Humans: ÷ Computers: (✓)

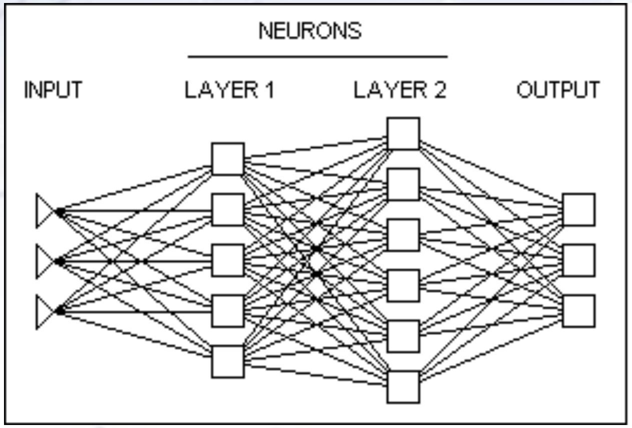
Computers, on the other hand, are OK with high dimensionality, albeit the growth of the challenge, but have a harder time facing non-linear issues.

However, through smart algorithms, computers have learned to deal with it all!
That is essentially what Machine Learning has enabled!

Data Mining

Seeing patterns in data and using it!



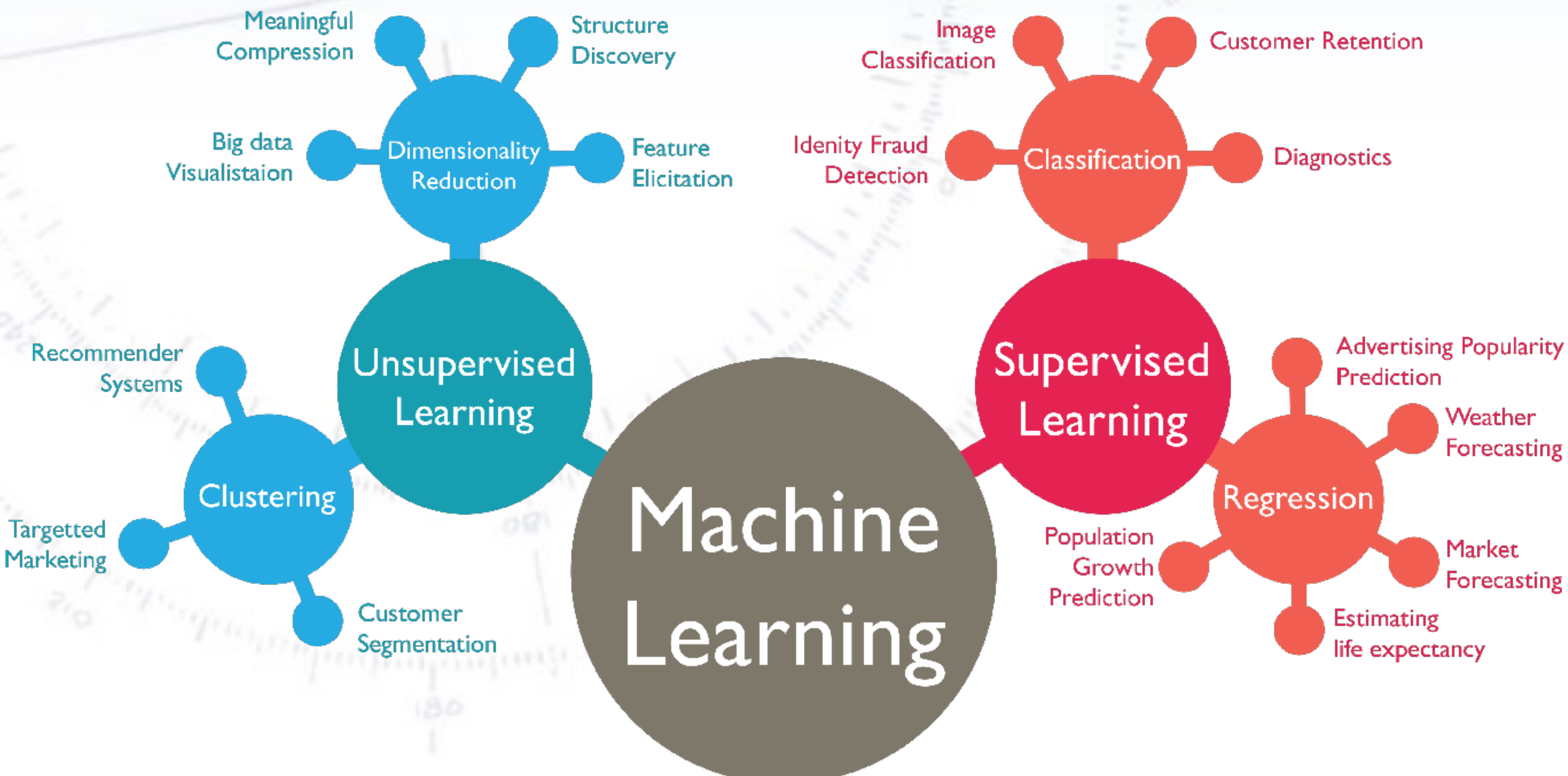


Data mining** is the process of extracting patterns from data. As more data are gathered, with the amount of data doubling every three years, data mining is becoming an increasingly important tool to transform these data into information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and **scientific discovery.

[Wikipedia, Introduction to Data Mining]

Unsupervised vs. Supervised
Classification vs. Regression

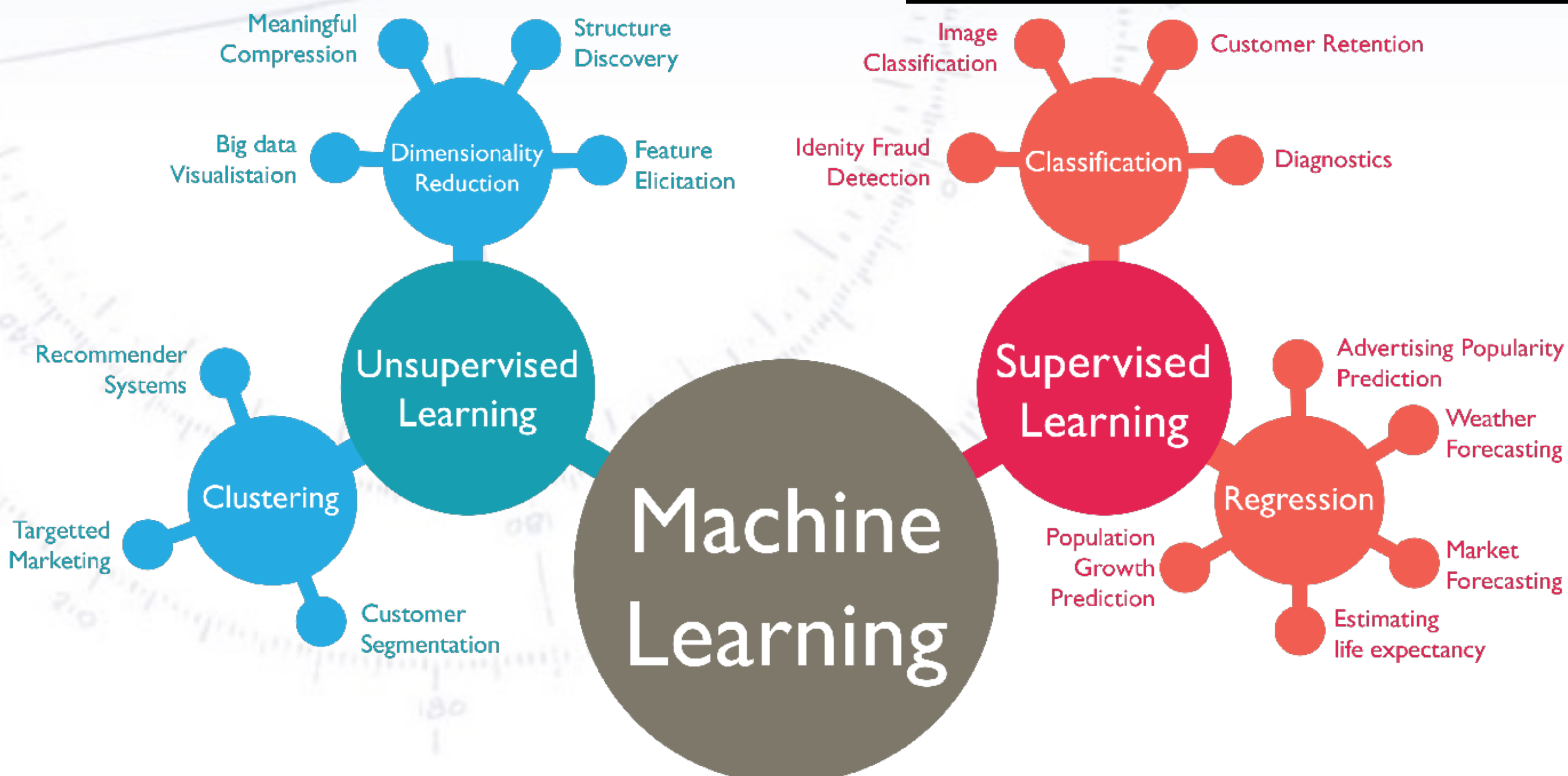
Machine Learning can be supervised (you have correctly labelled examples) or unsupervised (you don't)… [or reinforced]. Following this, one can be using ML to either classify (is it A or B?) or for regression (estimate of X).



Unsupervised vs. Supervised Classification vs. Regression

Machine Learning can be supervised (you have correctly labelled examples) or unsupervised (you don't)… [or reinforced]. Following this, one can be using ML to either classify (is it A or B?) or for regression (estimate of X).

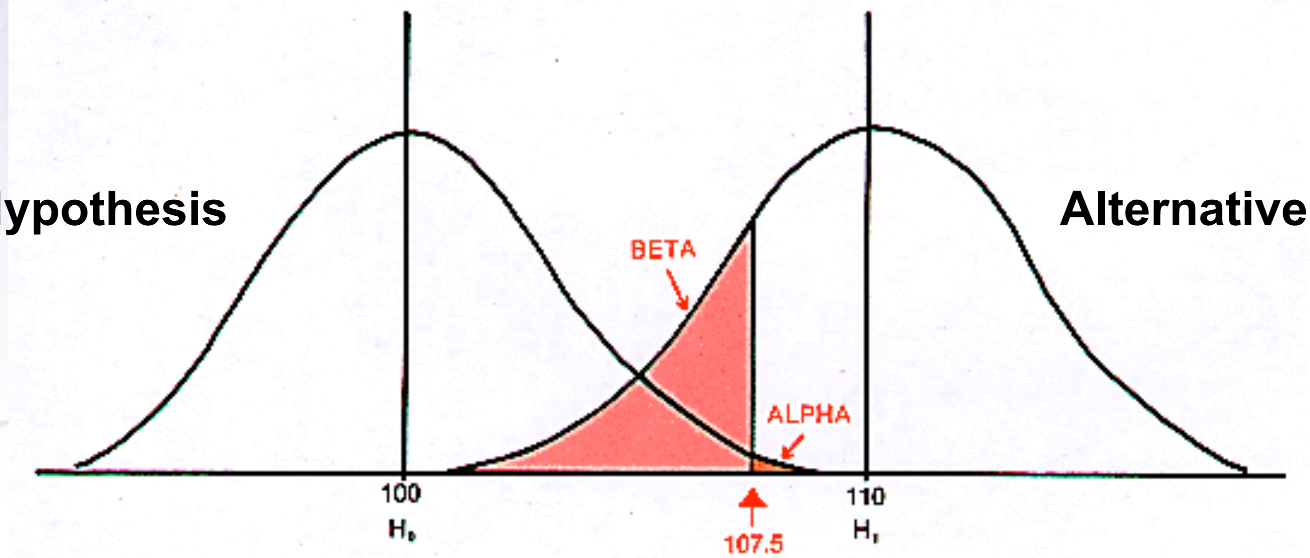
We will be mostly on this side!



Classification/Hypothesis

Null Hypothesis

Alternative Hypothesis



STATISTICAL DECISION:

		REALITY	
		Null is True	Null is False
STATISTICAL DECISION:	Do Not Reject Null	$1-\alpha$ Correct	β Type II error
	Reject Null	α Type I error	$1-\beta$ Correct

Hypothesis testing

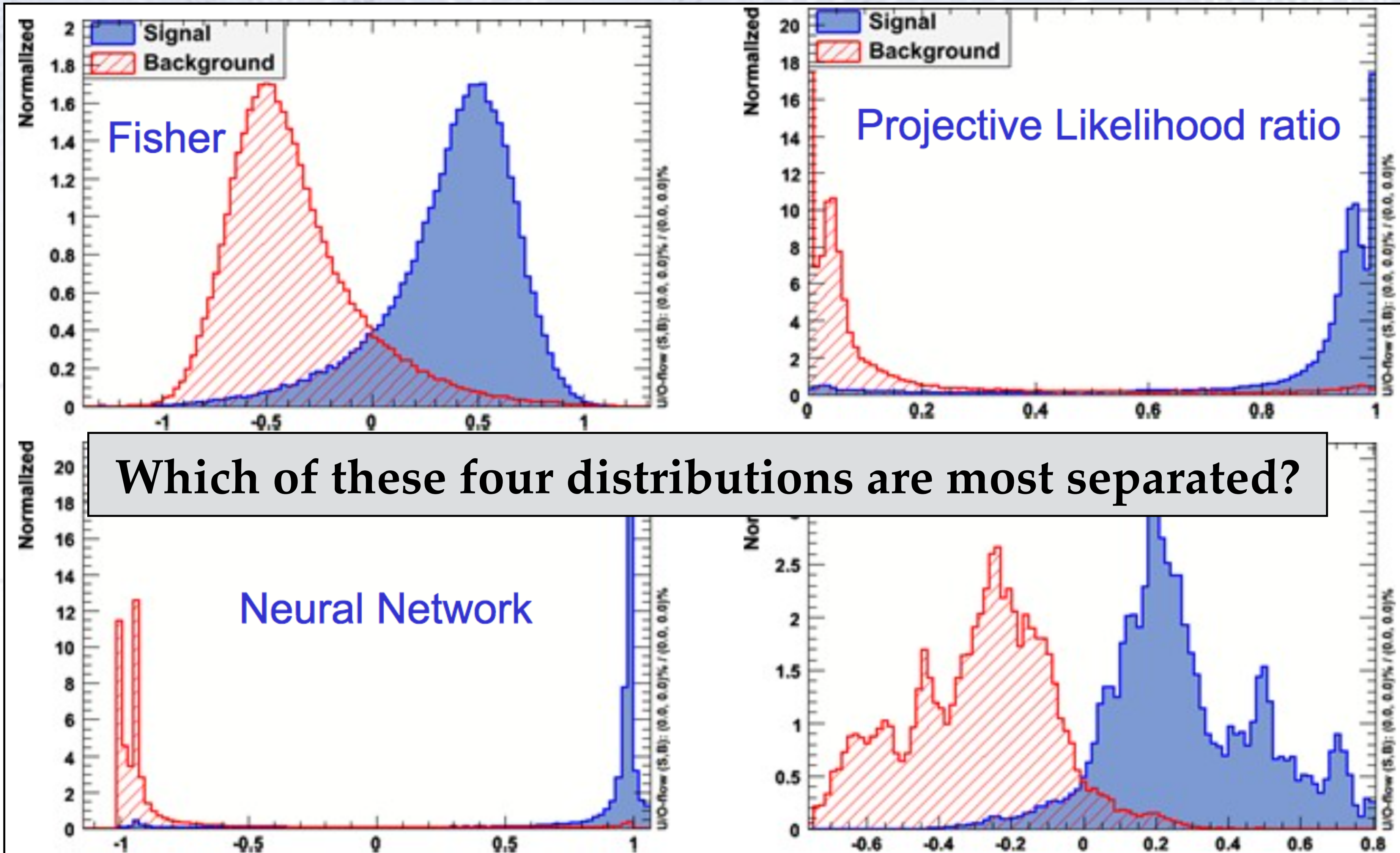
Hypothesis testing is like a criminal trial. The basic "null" hypothesis is **Innocent** (called H_0) and this is the hypothesis we want to test, compared to an "alternative" hypothesis, **Guilty** (called H_1).

Innocence is initially assumed, and this hypothesis is only rejected, if enough evidence proves otherwise, i.e. that the probability of innocence is very small ("beyond reasonable doubt").

	Truly innocent (H_0 is true)	**Truly guilty (H_1 is true)**
Acquittal (Accept H_0)	**Right decision**	**Wrong decision** **Type II error**
Conviction (Reject H_0)	**Wrong decision** **Type I error**	**Right decision**

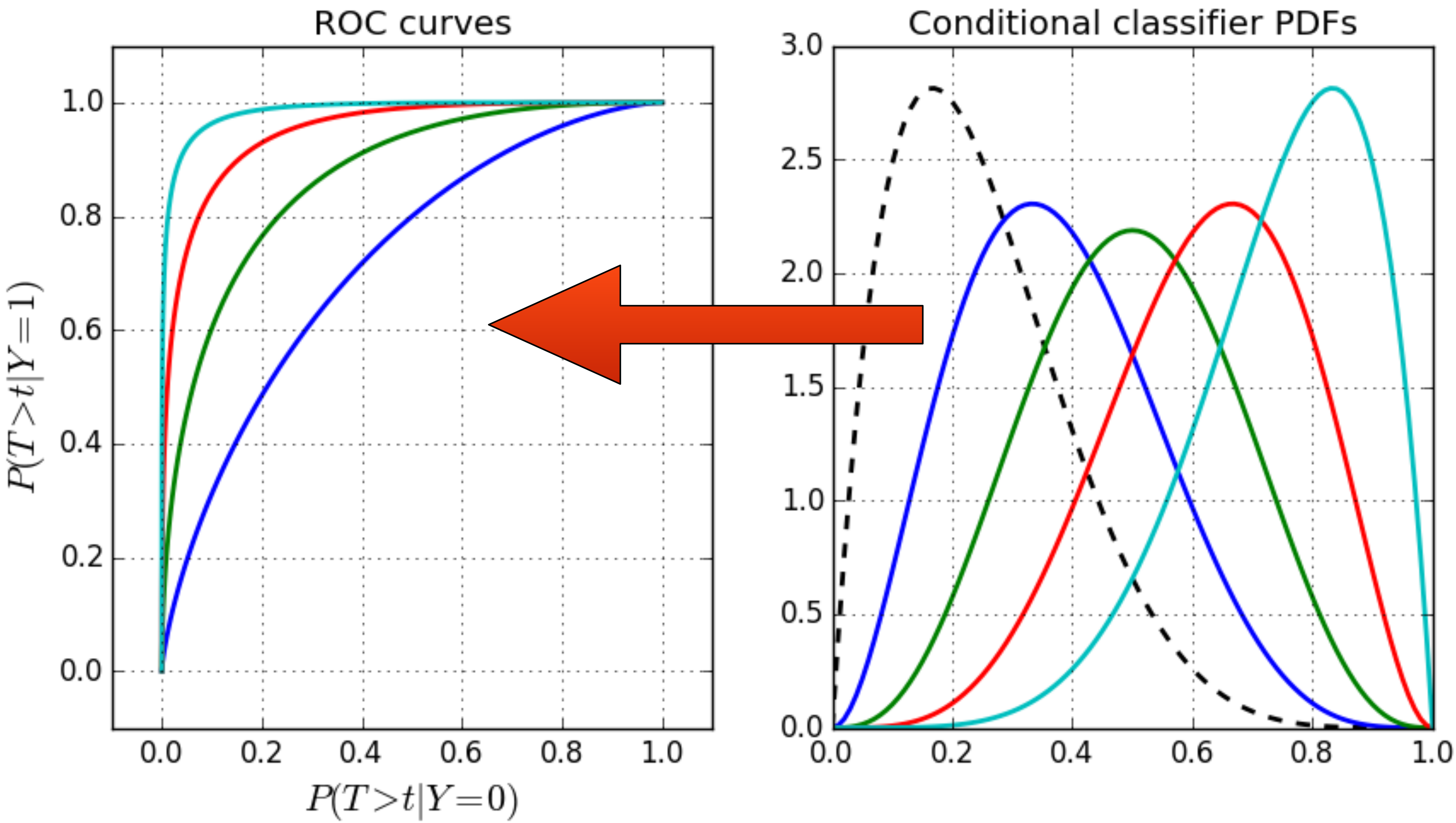
The rate of type I/II errors are correlated, and one can only choose one of these!

Measuring separation



Which of these four distributions are most separated?

Simple case

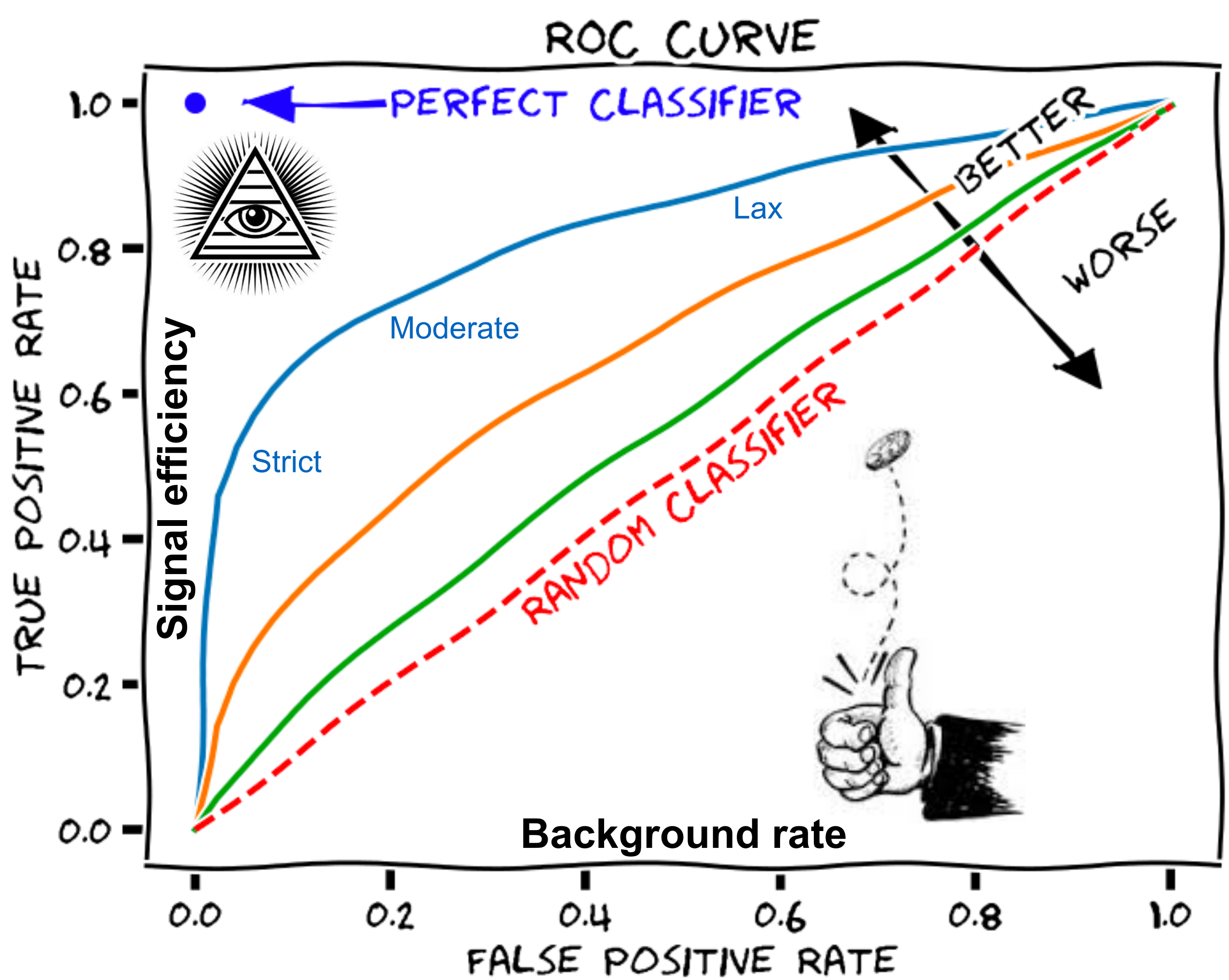

ROC curves
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
$P(T>t|Y=1)$
$P(T>t|Y=0)$
Conditional classifier PDFs
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.2
0.4
0.6
0.8
1.0

ROC curves

A **Receiver Operating Characteristic** or just ROC-curve is a graphical plot of the sensitivity, or true positive rate (TPR), vs. false positive rate (FPR).

It is calculated as the integral of the two hypothesis distributions, and is used to evaluate a test.

Given classification into two distributions, one may choose to make a strict or lax selection.
This choice depends on the case at hand.



Which metric to use?

There are a ton of metrics in hypothesis testing, see below. However, those in the boxes below are the most central ones.

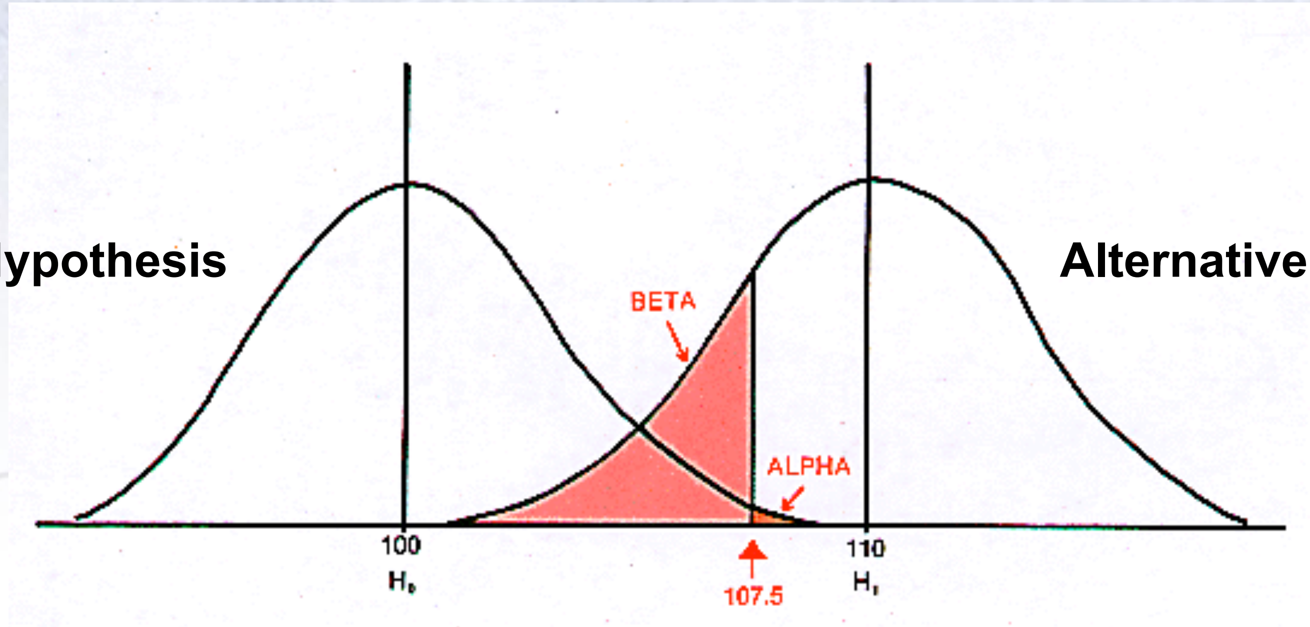
One metric - not mentioned here - is the Area Under the Curve (AUC), which is simply an integral of the ROC curve (thus 1 is perfect score). This is often used in Machine Learning to optimise performance (loss).

		True condition				
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
Predicted condition	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$	F_1 score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$		

https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Classification/Hypothesis

Null Hypothesis



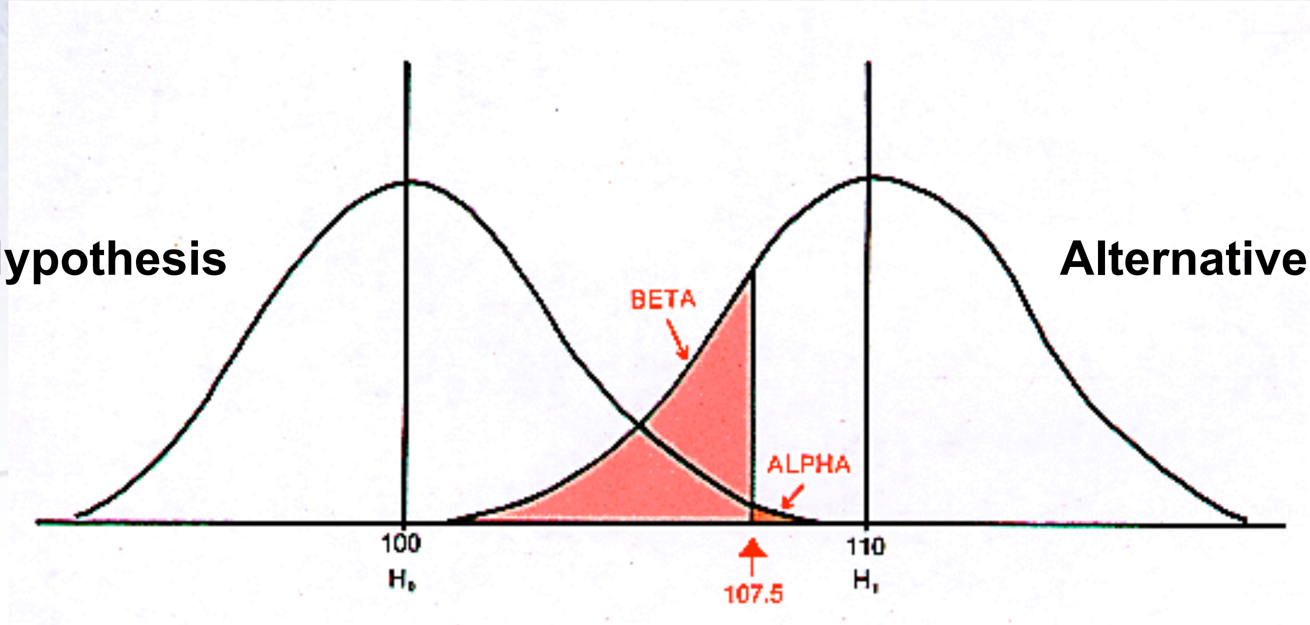
Alternative Hypothesis

		REALITY	
		Null is True	Null is False
STATISTICAL DECISION:	Do Not Reject Null	$1-\alpha$ Correct	β Type II error
	Reject Null	α Type I error	$1-\beta$ Correct

Classification/Hypothesis

Null Hypothesis

Alternative Hypothesis



Machine Learning typically enables a better separation between hypothesis

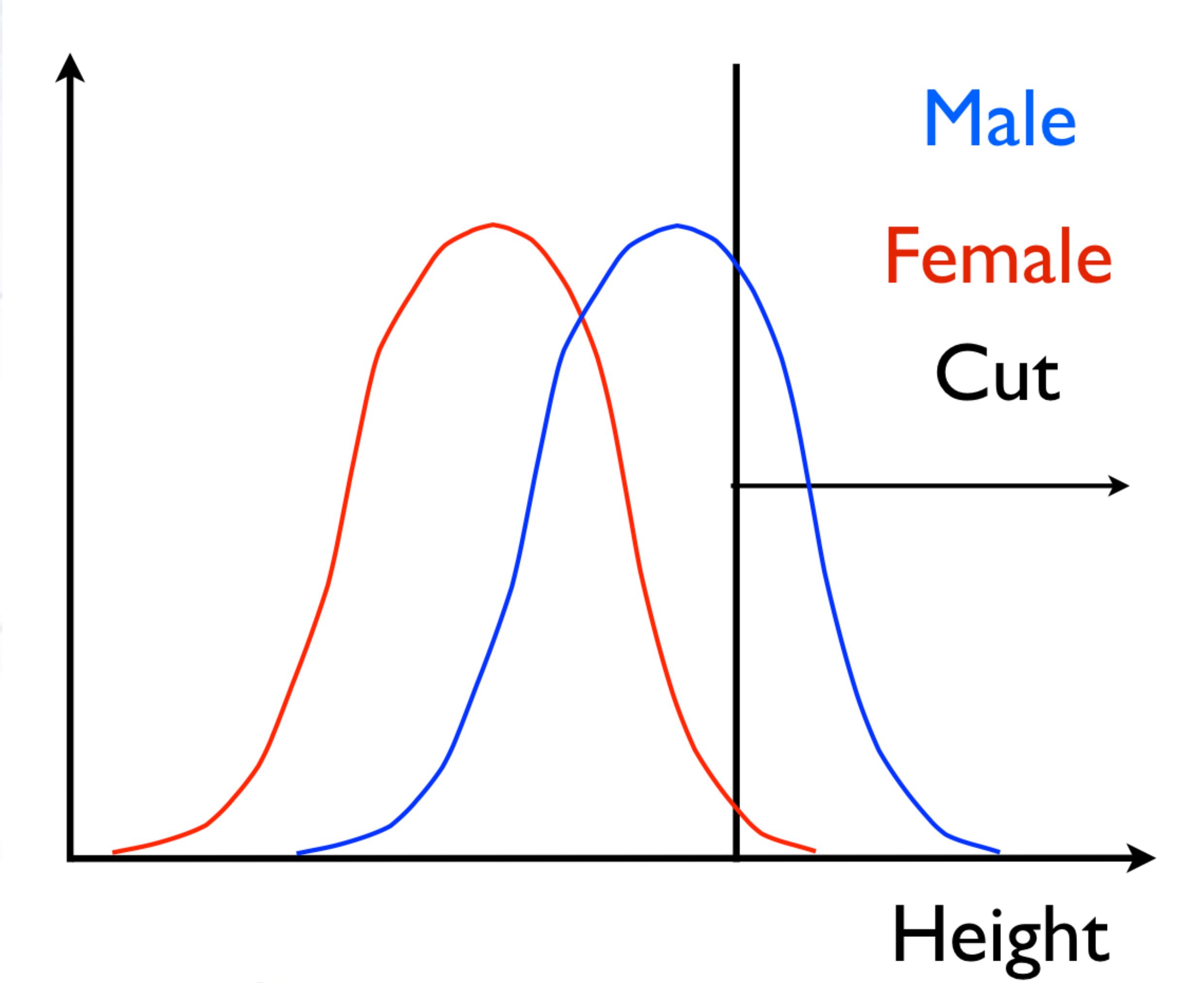
DECISION:

Reject Null	α Type I error	1 – β Correct

The linear case

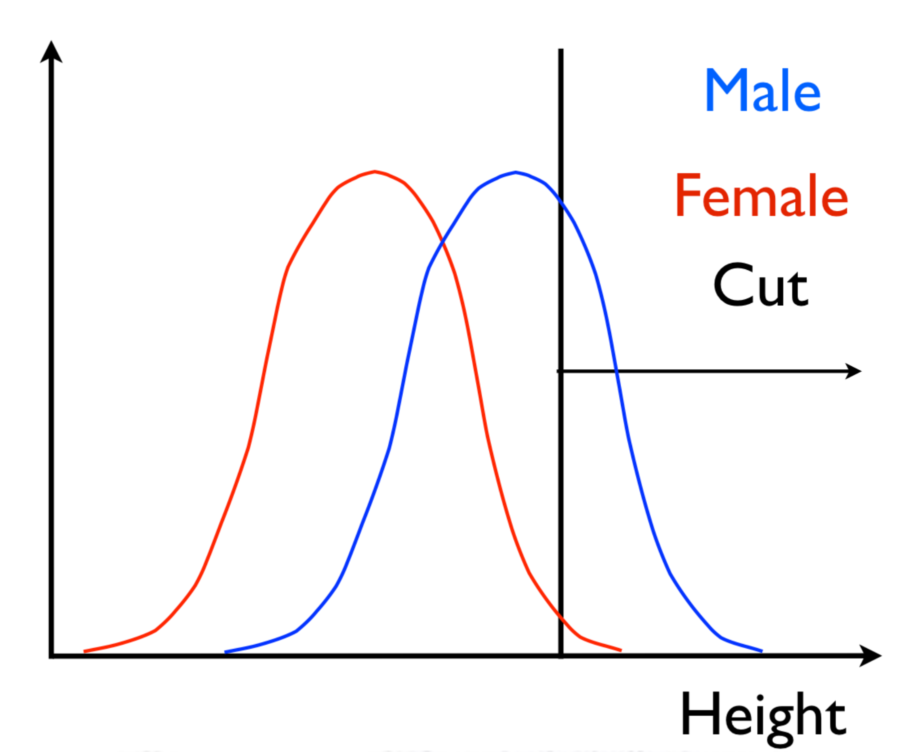
Simple Example

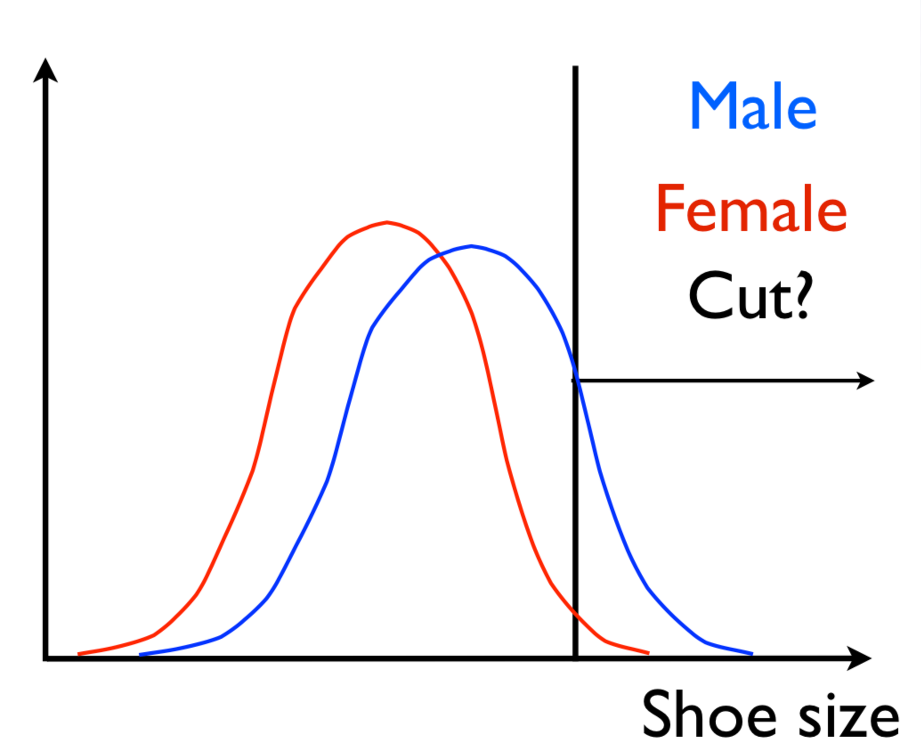
Problem: You want to figure out a method for getting sample that is mostly male!
Solution: Gather height data from 10000 people, Estimate cut with 95% purity!



Simple Example

Additional data: The data you find also contains shoe size!
How to use this? Well, it is more information, but should you cut on it?



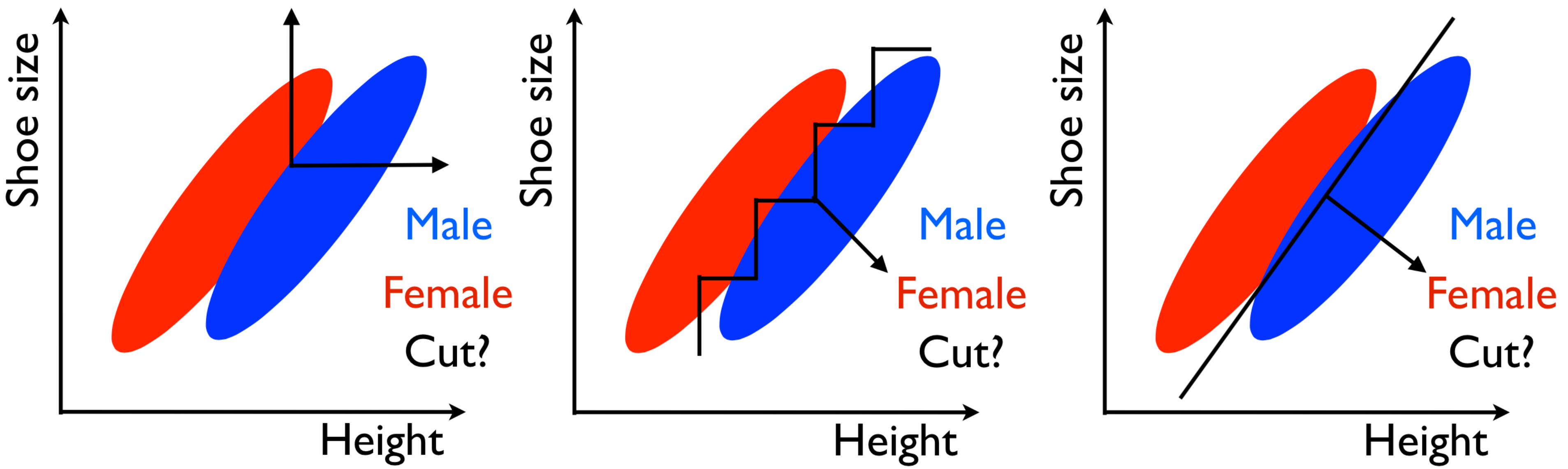


The question is, what is the best way to use this (possibly correlated) information!

Simple Example

So we look if the data is correlated, and consider the options:

Cut on each var?	Advanced cut?	Combine var?
Poor efficiency!	Clumsy and hard to implement	Smart and promising



The latter approach is the Fisher discriminant!
It has the advantage of being simple and applicable in many dimensions easily!

Separating data

Fisher's friend, Anderson, came home from picking Irises in the Gaspe peninsula...

180 MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS

Table I

Iris setosa				*Iris versicolor*				*Iris virginica*			
Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
5·1	3·5	1·4	0·2	7·0	3·2	4·7	1·4	6·3	3·3	6·0	2·5
4·9	3·0	1·4	0·2	6·4	3·2	4·5	1·5	5·8	2·7	5·1	1·9
4·7	3·2	1·3	0·2	6·9	3·1	4·9	1·5	7·1	3·0	5·9	2·1
4·6	3·1	1·5	0·2	5·5	2·3	4·0	1·3	6·3	2·9	5·6	1·8
5·8	4·0	1·2	0·2	5·6	2·9	3·6	1·3	5·8	2·8	5·1	2·4
5·7	4·4	1·5	0·4	6·7	3·1	4·4	1·4	6·4	3·2	5·3	2·3
5·4	3·9	1·3	0·4	5·6	3·0	4·5	1·5	6·5	3·0	5·5	1·8
5·1	3·5	1·4	0·3	5·8	2·7	4·1	1·0	7·7	3·8	6·7	2·2
5·7	3·8	1·7	0·3	6·2	2·2	4·5	1·5	7·7	2·6	6·9	2·3

Fisher’s Linear Discriminant

You want to separate two types/classes (A and B) of events using several measurements.

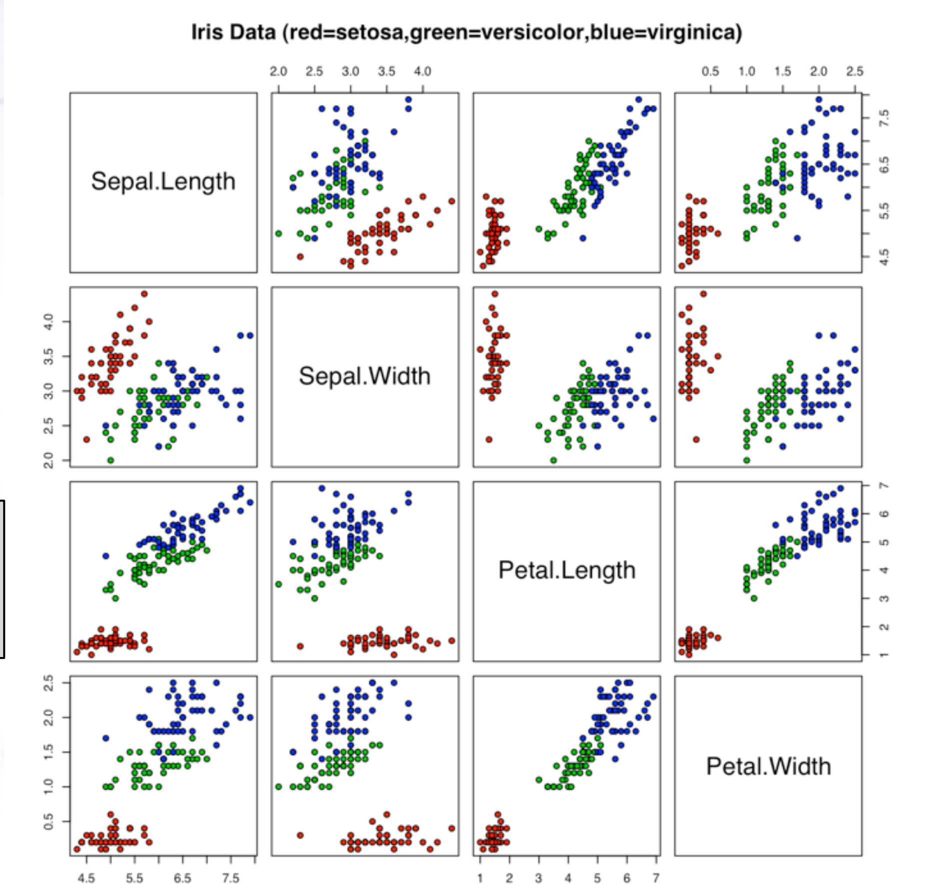
Q: How to combine the variables?
A: Use the Fisher Discriminant:

$$\mathcal{F} = w_0 + \vec{w} \cdot \vec{x}$$

Q: How to choose the values of w?
A: Inverting the covariance matrices:

$$\vec{w} = (\mathbf{\Sigma}_A + \mathbf{\Sigma}_B)^{-1} \ (\vec{\mu}_A - \vec{\mu}_B)$$

This can be calculated analytically, and incorporates the linear correlations into the separation capability.



Fisher's Linear Discriminant

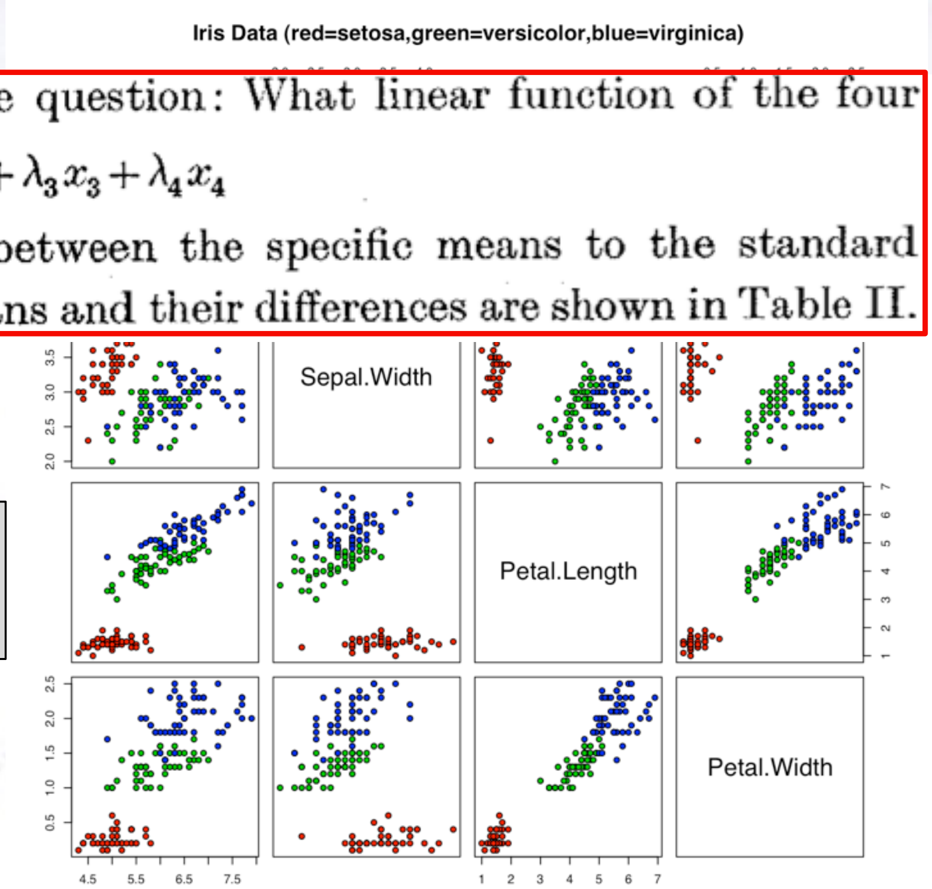
You want to separate two types/classes (A and B) of events using several measurements.

<u>Q</u>: How to combine the variables?
A: Use the Fisher Discriminant:

ments are given. We shall first consider the question: What linear function of the four measurements

$$X = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + \lambda_4 x_4$$

will maximize the ratio of the difference between the specific means to the standard deviations within species? The observed means and their differences are shown in Table II.



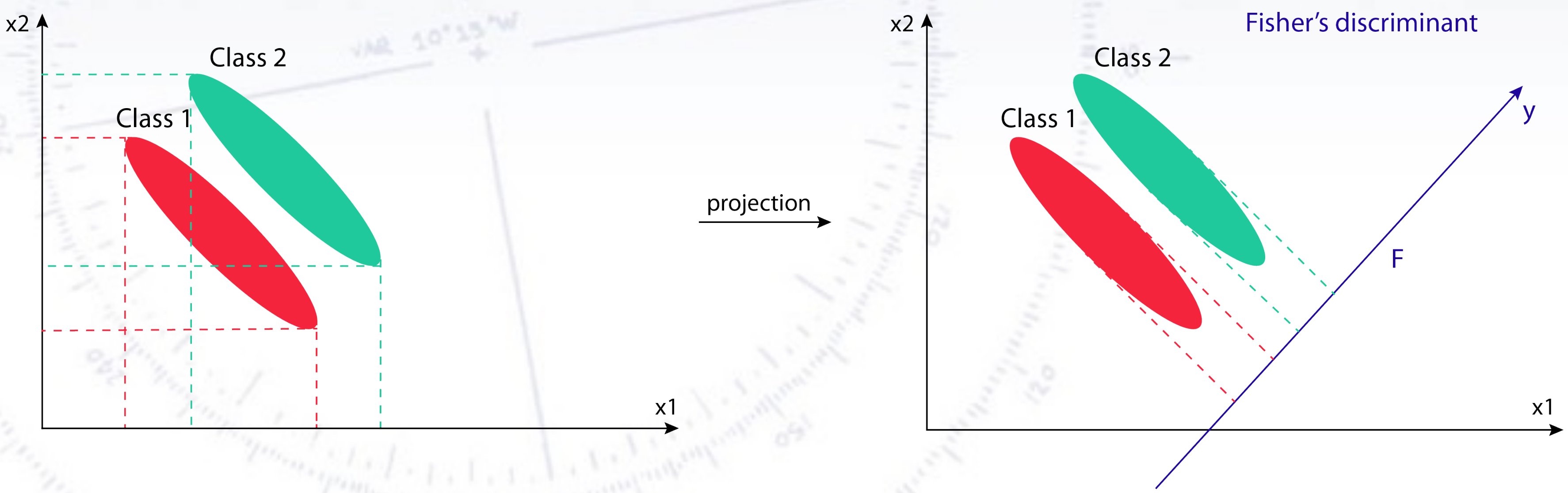
<u>Q</u>: How to choose the values of w?
<u>A</u>: Inverting the covariance matrices:

$$\vec{w} = (\mathbf{\Sigma}_A + \mathbf{\Sigma}_B)^{-1} \ (\vec{\mu}_A - \vec{\mu}_B)$$

This can be calculated analytically, and incorporates the linear correlations into the separation capability.

Fisher's Linear Discriminant

<u>Executive summary:</u>
Fisher's Discriminant uses a linear combination of variables to give a single variable with the maximum possible separation (for linear combinations!).



It is for all practical purposes a projection (in a Euclidian space)!

Fisher's Linear Discriminant

The details of the formula are outlined below:

You have two samples, A and B, that you want to separate.

For each input variable (x), you calculate the mean (μ), and form a vector of these.

$$\vec{w} = \left(\boldsymbol{\Sigma}_A + \boldsymbol{\Sigma}_B\right)^{-1} \left(\vec{\mu}_A - \vec{\mu}_B\right)$$

Using the input variables (x), you calculate the covariance matrix (Σ) for each species (A/B), add these and invert.

Given weights (w), you take your input variables (x) and combine them linearly as follows:

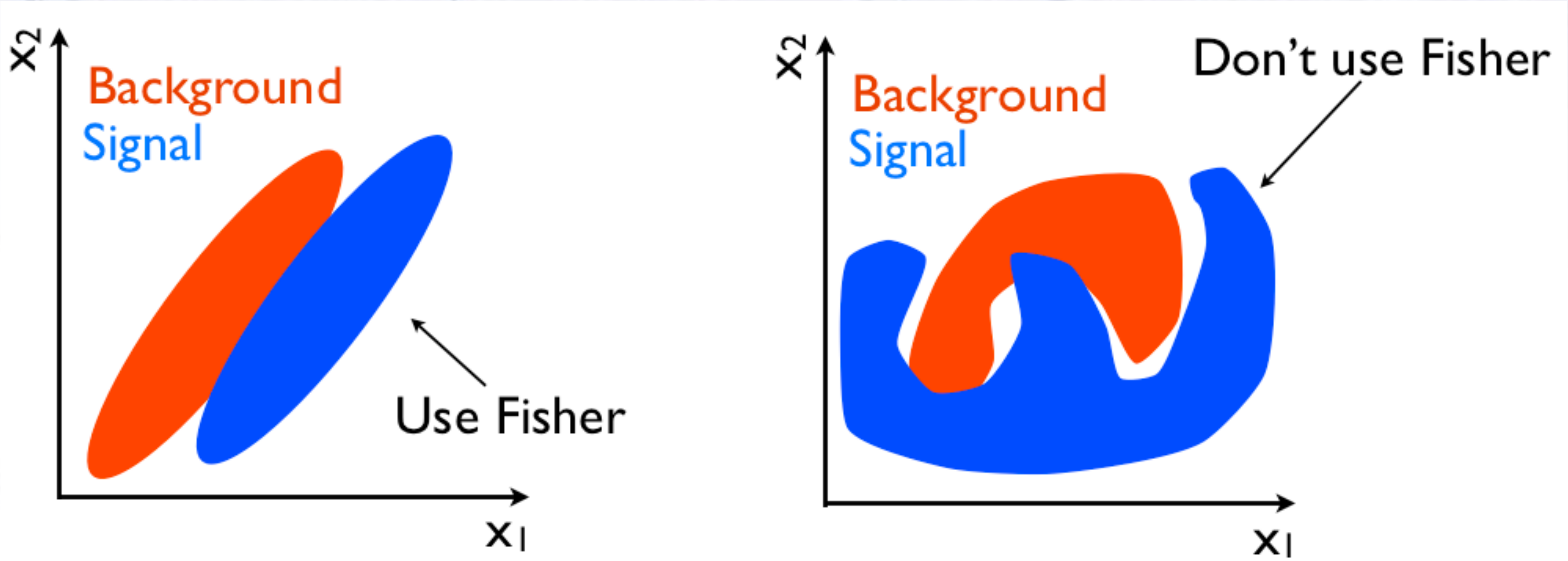
$$\mathcal{F} = w_0 + \vec{w} \cdot \vec{x}$$

F is what you base your decision on.

The non-linear case

Non-linear cases

While the Fisher Discriminant uses all separations and **linear correlations**, it does not perform optimally, when there are **non-linear correlations** present:



If the PDFs of signal and background are known, then one can use a likelihood. But this is **very rarely** the case, and hence one should move on to the Fisher. However, if correlations are non-linear, more "tough" methods are needed...

Neural Networks

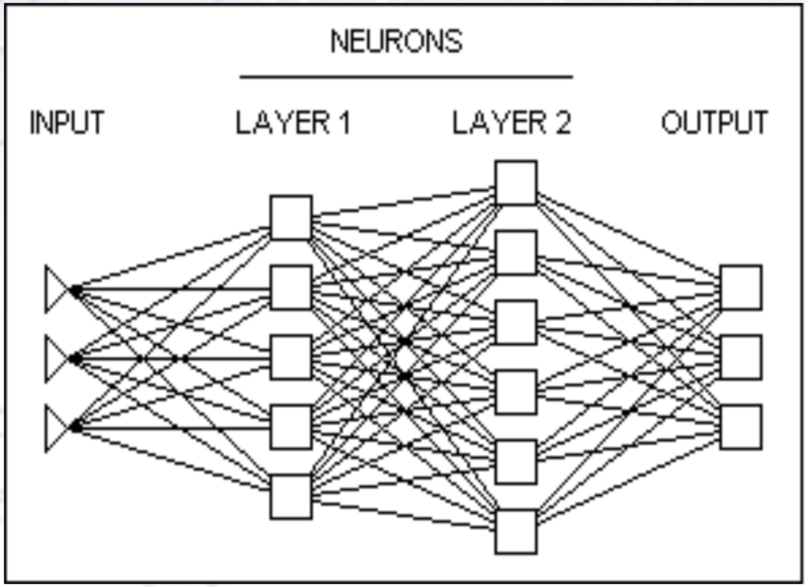
Can become very complex.

Good for continuous problems.

Sometimes hard to train!

Can be used for images.

Easily produces multiple outputs.



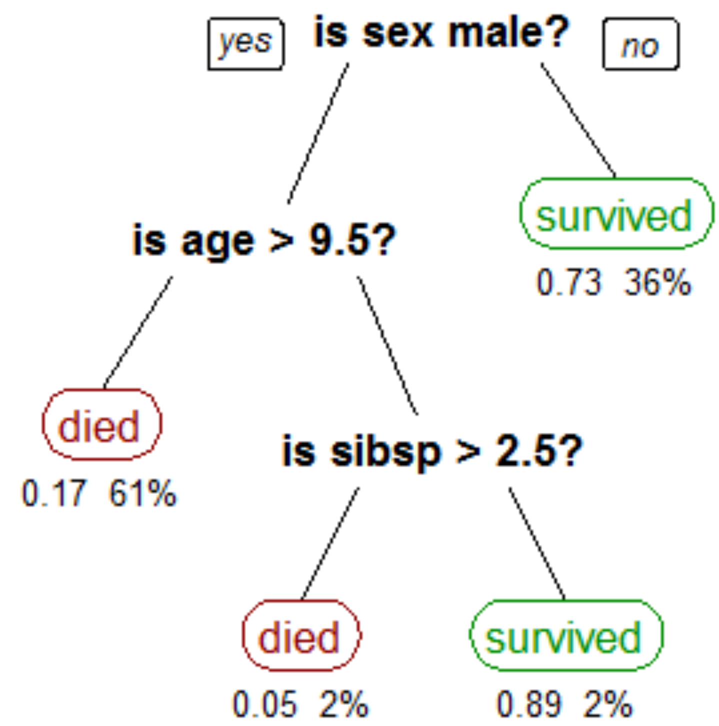
(Boosted) Decision Trees

Can become very complex.

Good for discrete problems.
"Good for all problems!!!"

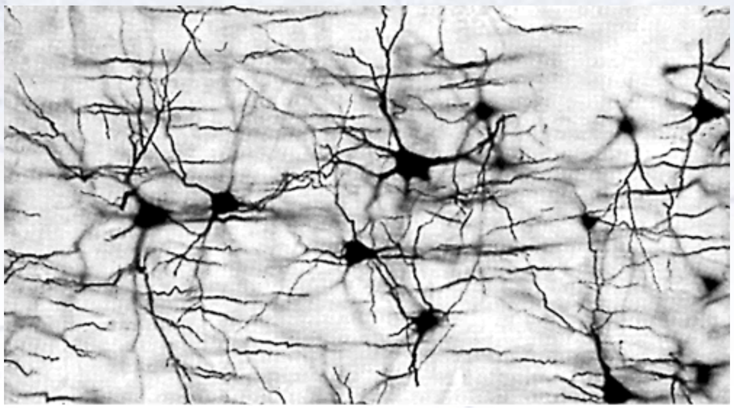
Not always highest efficiency.

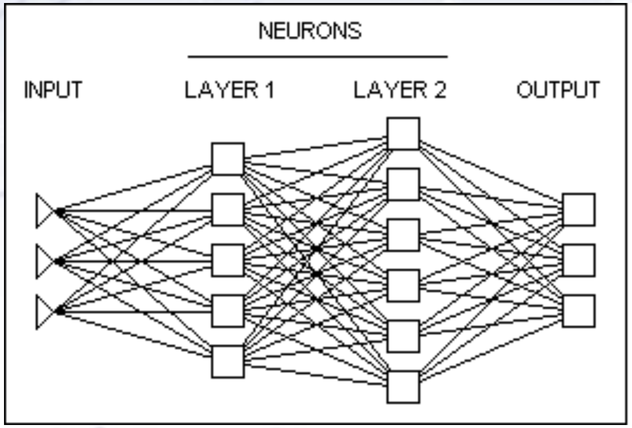
Boosting adds to separation.



* The example BDT shown is a simple example for predicting survival of Titanic!

Neural Networks (NN)





In machine learning and related fields, artificial neural networks (ANNs) are computational models inspired by an animal's central nervous systems (in particular the brain) which is capable of ***machine learning*** *as well as* ***pattern recognition****.*

Neural networks *have been used to solve a wide variety of tasks that are hard to solve using ordinary rule-based programming, including* ***computer vision*** *and* ***speech recognition****.*

[Wikipedia, Introduction to Artificial Neural Network]

Neural Networks

Neural Networks combine the input variables using a "activation" function s(x) to assign, if the variable indicates signal or background.

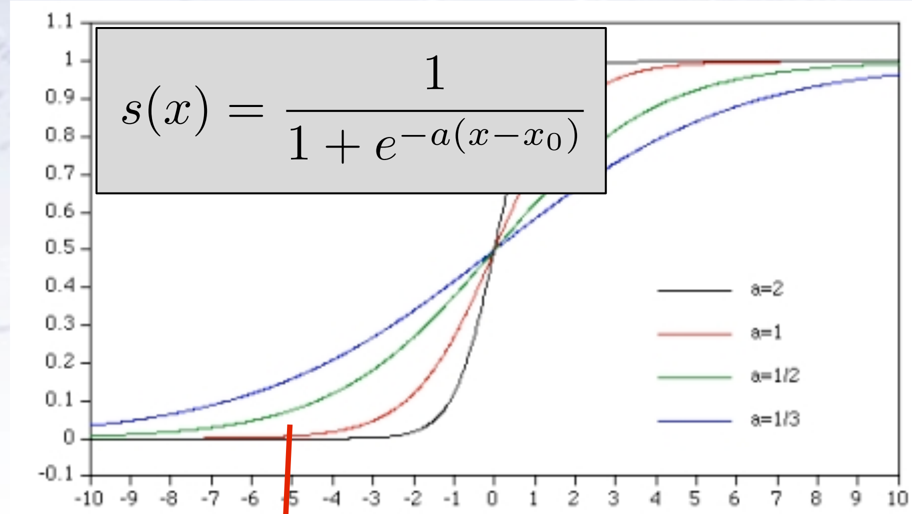
The simplest is a single layer perceptron:

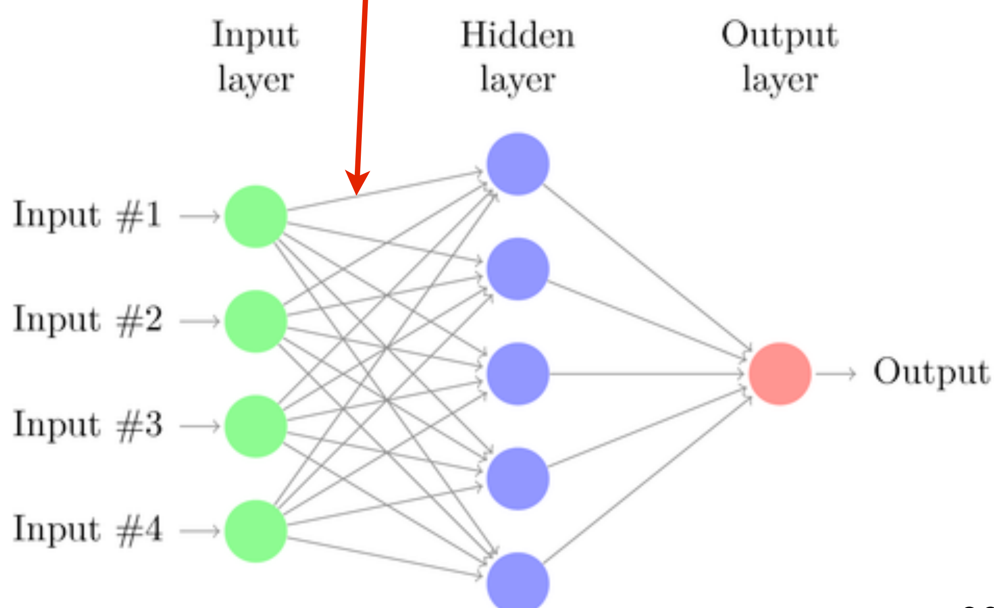
$$t(x) = s\left(a_0 + \sum a_i x_i\right)$$

This can be generalised to a multilayer perceptron:

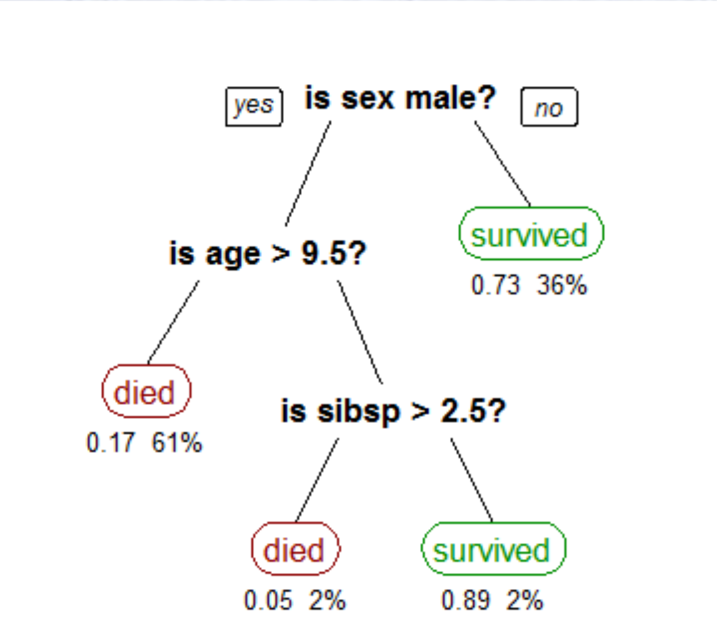
$$t(x) = s\left(a_i + \sum a_i h_i(x)\right)$$
$$h_i(x) = s\left(w_{i0} + \sum w_{ij} x_j\right)$$

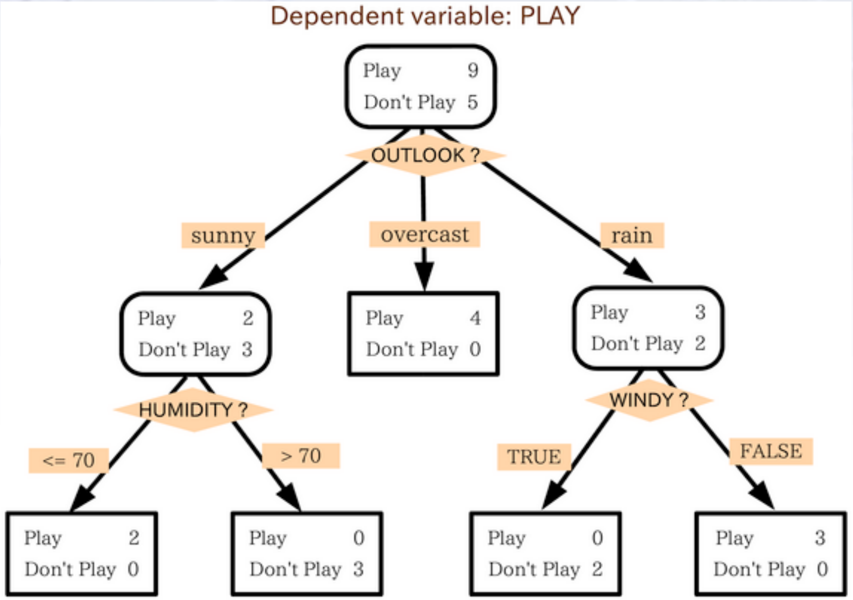
Activation function can be any sigmoid function.





Boosted Decision Trees (BDT)





__Decision tree learning__ uses a __decision tree__ as a __predictive model__ which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in __statistics__, __data mining__ and __machine learning__.

[Wikipedia, Introduction to Decision Tree Learning]

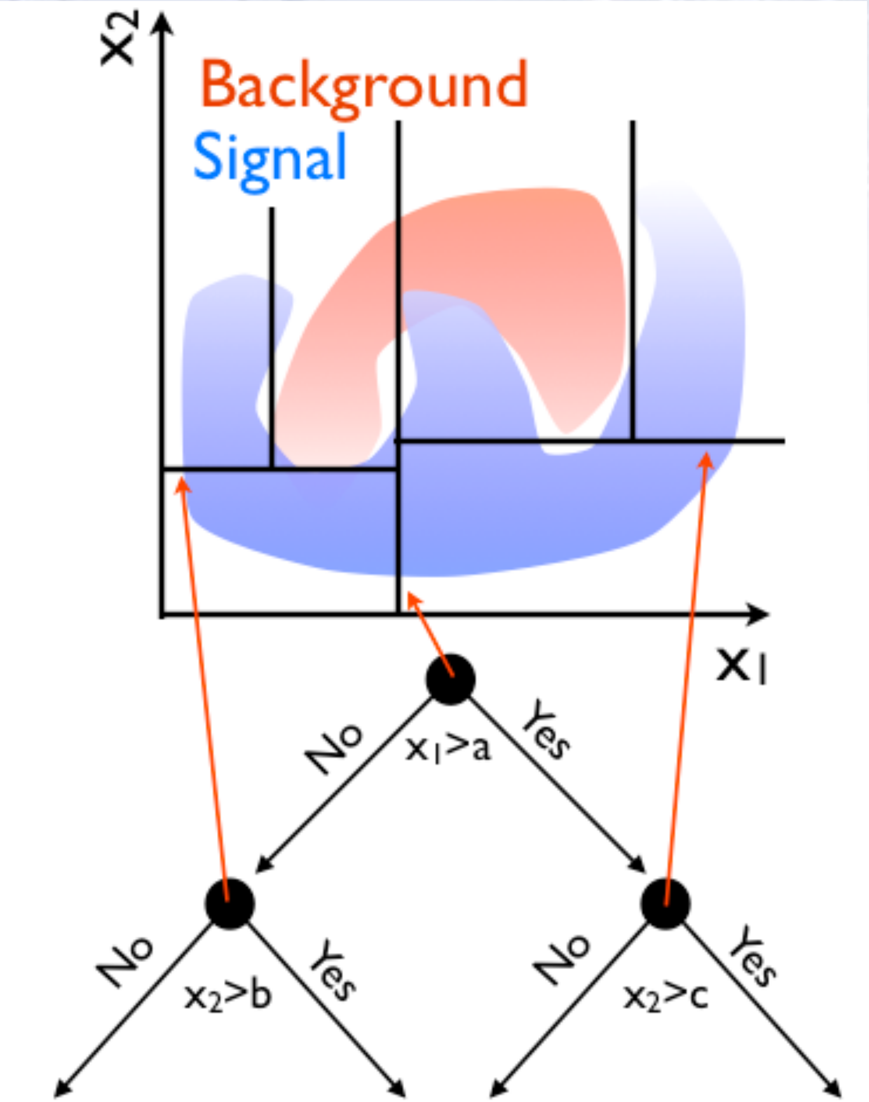
Boosted Decision Trees

A decision tree divides the parameter space, starting with the maximal separation. In the end each part has a probability of being signal or background.

- Works in 95+% of all problems!
- Fully uses non-linear correlations.

But BDTs require a lot of data for training, and is sensitive to overtraining (see next slide).

Overtraining can be reduced by limiting the number of nodes and number of trees.



Boosting...

There is no reason, why you can not have more trees. Each tree is a simple classifier, but many can be combined!

To avoid N identical trees, one assigns a higher weight to events that are hard to classify, i.e. boosting:

Boost weight

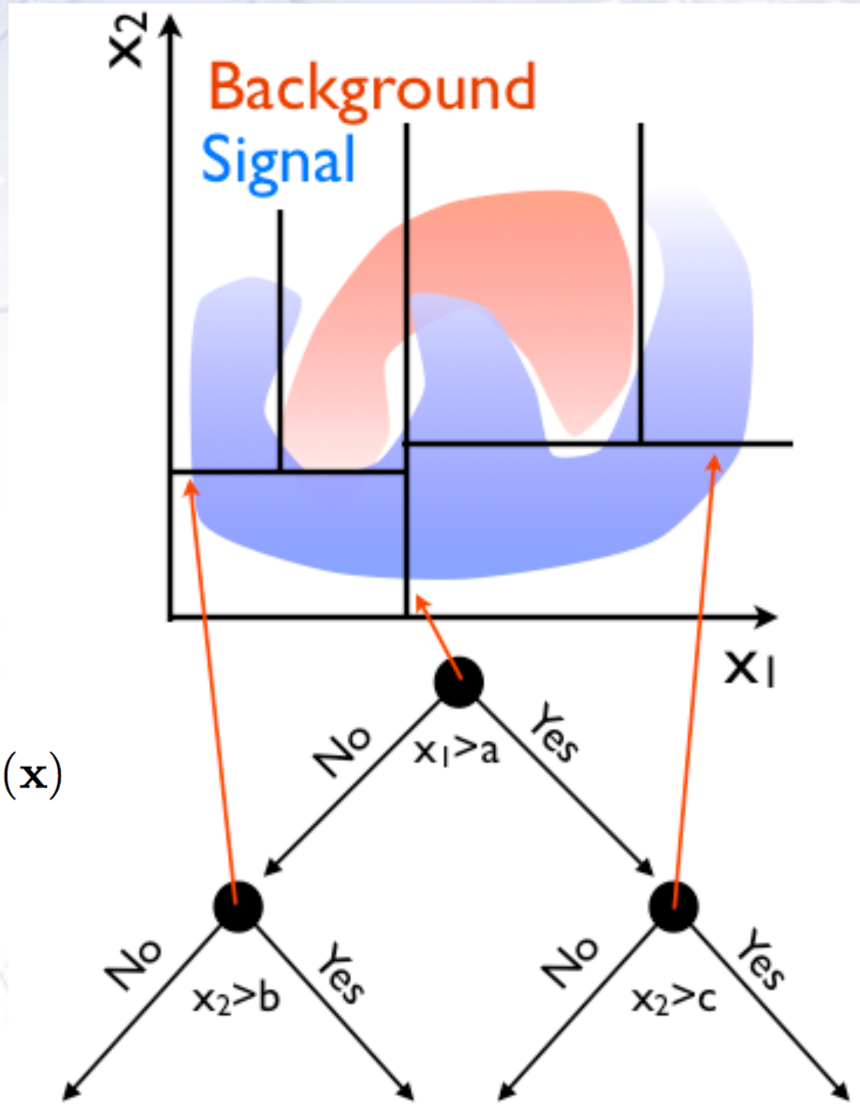
$$\alpha = \frac{1 - \mathrm{err}}{\mathrm{err}}$$

First classifier

$$y_{\mathrm{Boost}}(\mathbf{x}) = \frac{1}{N_{\mathrm{collection}}} \cdot \sum_{i}^{N_{\mathrm{collection}}} \ln(\alpha_i) \cdot h_i(\mathbf{x})$$

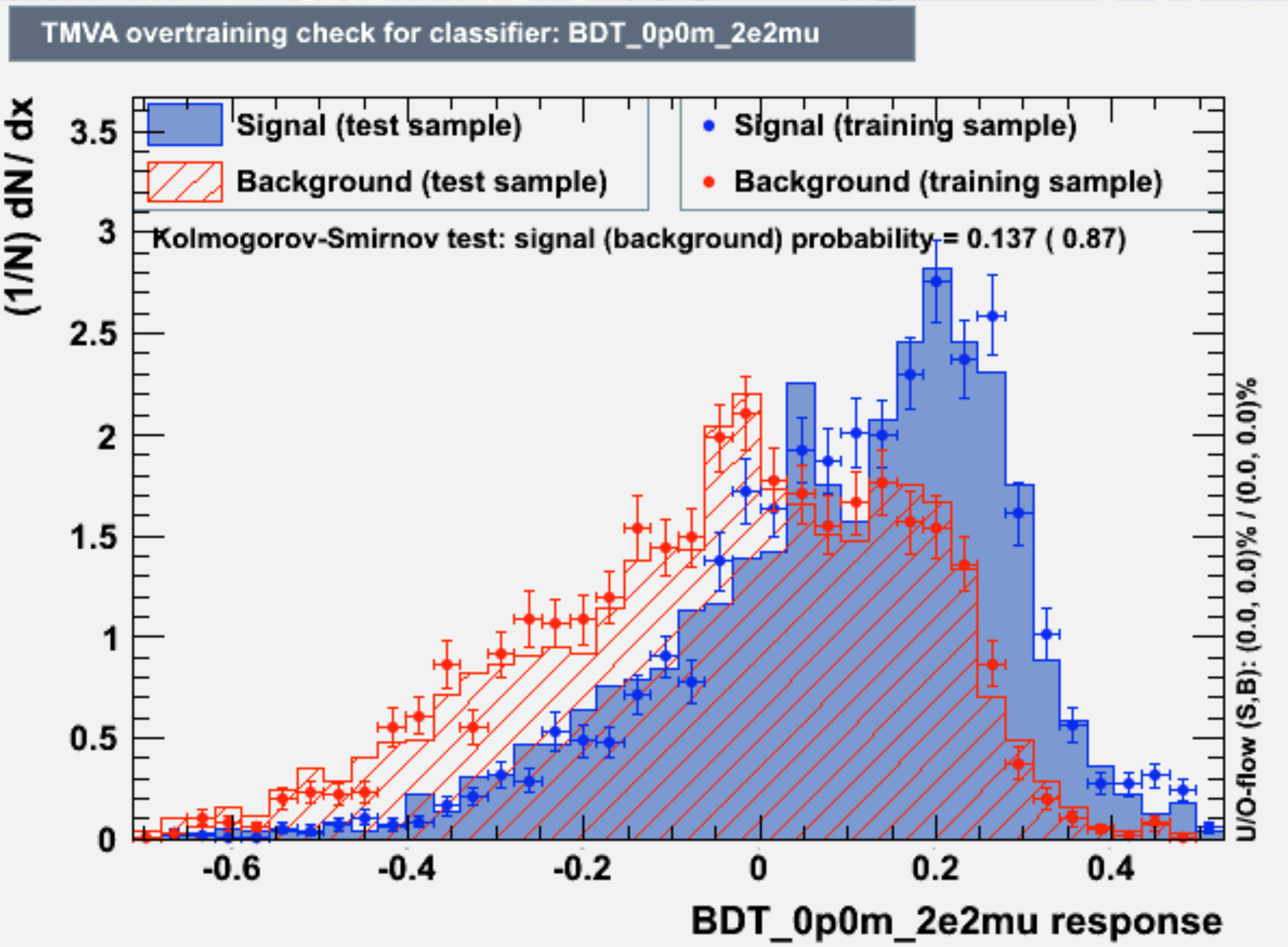
Parameters in event N

Individual tree



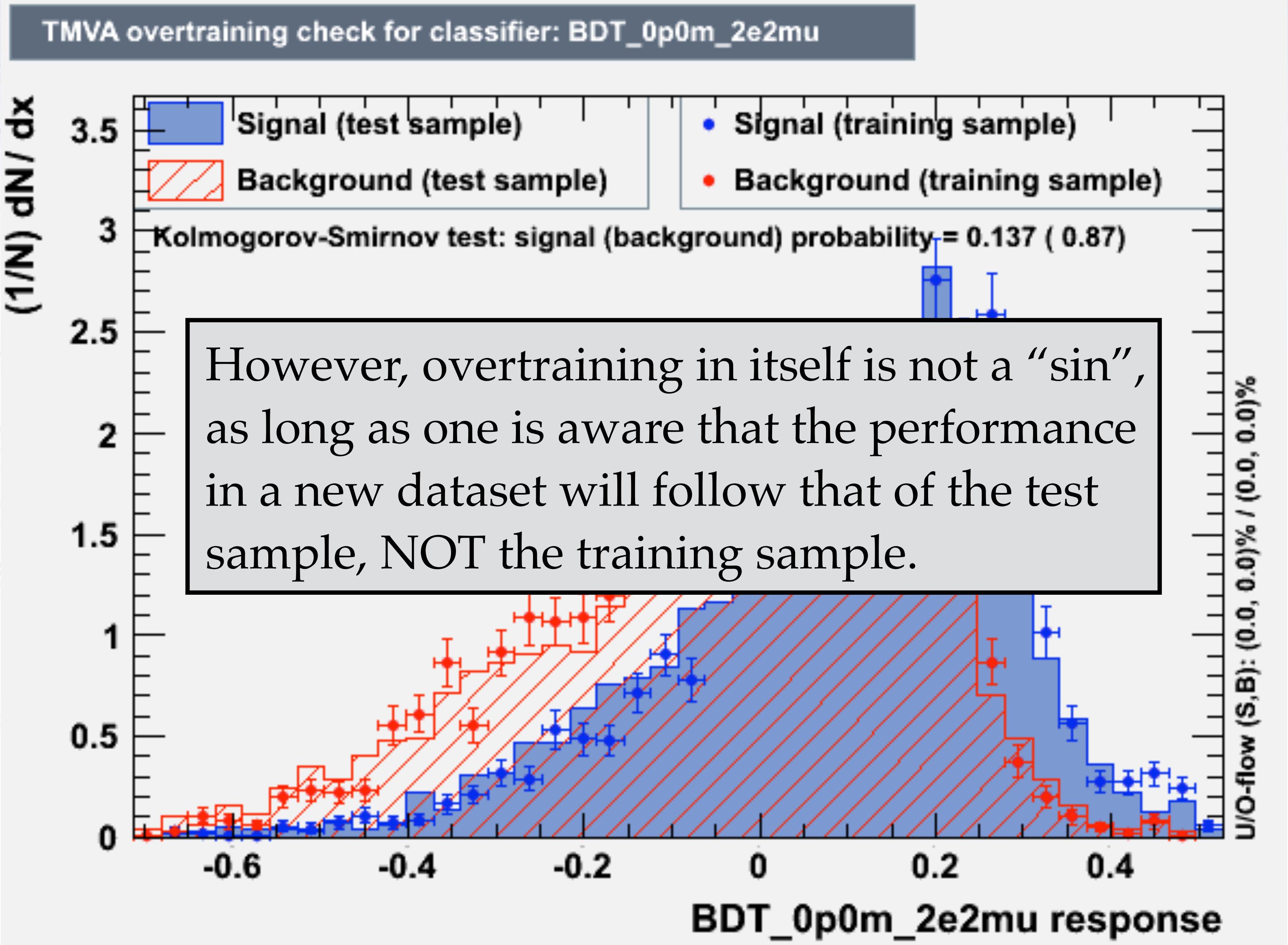
Test for simple overtraining

In order to test for overtraining, half the sample is used for training, the other for testing:



Test for simple overtraining

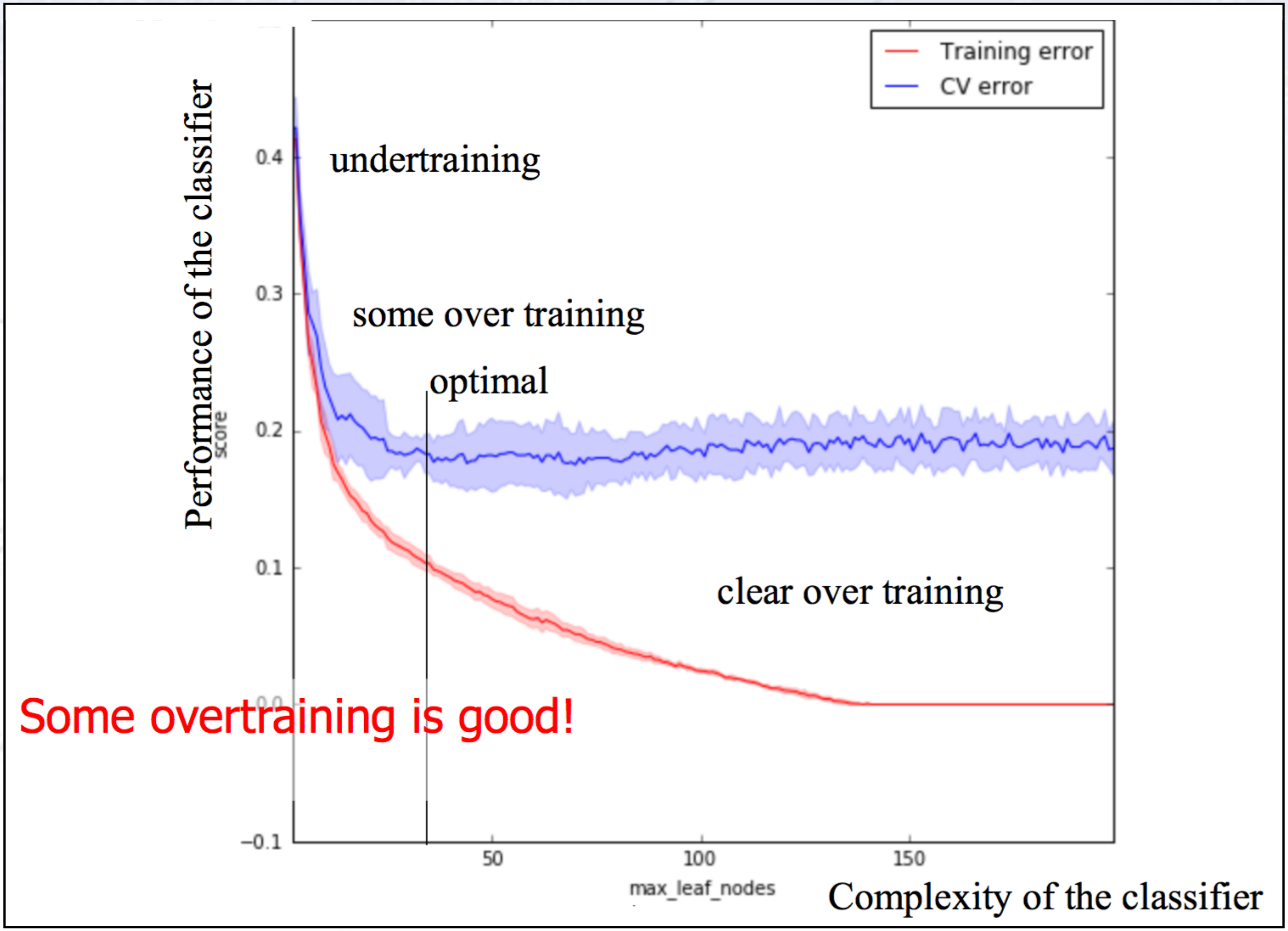
In order to test for overtraining, half the sample is used for training, the other for testing:



However, overtraining in itself is not a "sin", as long as one is aware that the performance in a new dataset will follow that of the test sample, NOT the training sample.

Real overtraining

The “real” limit of overtraining, is when the (Cross) Validation (CV) error starts to grow!



Method's (dis-)advantages

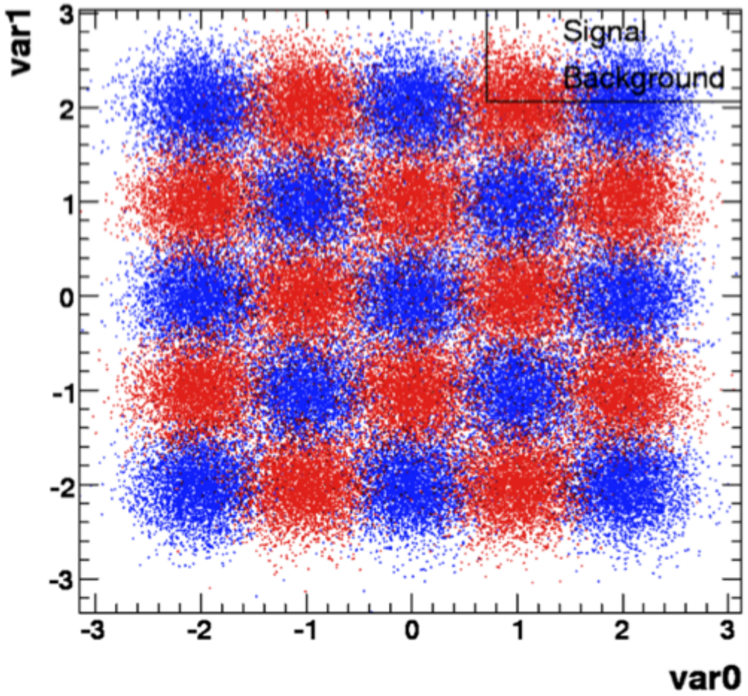
		CLASSIFIERS									
	CRITERIA	Cuts	Likeli-hood	PDE-RS	k-NN	H-Matrix	Fisher	ANN	BDT	Rule-Fit	SVM
Perfor-mance	No or linear correlations	⋆	⋆⋆	⋆	⋆	⋆	⋆⋆	⋆⋆	⋆	⋆⋆	⋆
	Nonlinear correlations	○	○	⋆⋆	⋆⋆	○	○	⋆⋆	⋆⋆	⋆⋆	⋆⋆
Speed	Training	○	⋆⋆	⋆⋆	⋆⋆	⋆⋆	⋆⋆	⋆	○	⋆	○
	Response	⋆⋆	⋆⋆	○	⋆	⋆⋆	⋆⋆	⋆⋆	⋆	⋆⋆	⋆
Robust-ness	Overtraining	⋆⋆	⋆	⋆	⋆	⋆⋆	⋆⋆	⋆	○	⋆	⋆⋆
	Weak variables	⋆⋆	⋆	○	○	⋆⋆	⋆⋆	⋆	⋆⋆	⋆	⋆
Curse of dimensionality		○	⋆⋆	○	○	⋆⋆	⋆⋆	⋆	⋆	⋆	
Transparency		⋆⋆	⋆⋆	⋆	⋆	⋆⋆	⋆⋆	○	○	○	○

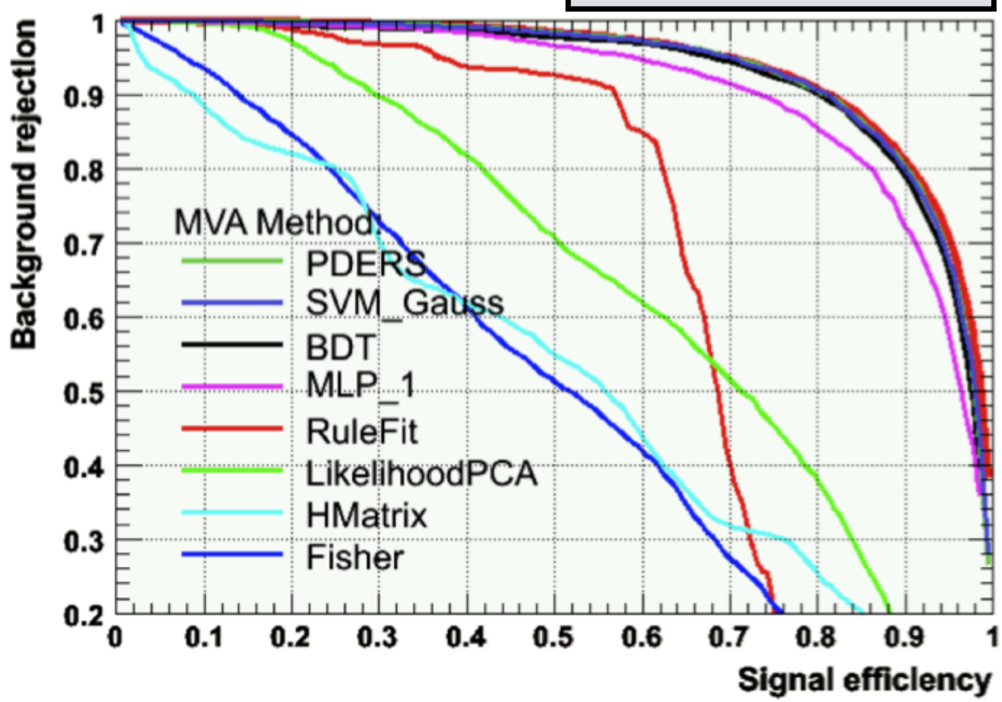
Table 1: Assessment of classifier properties. The symbols stand for the attributes "good" (⋆⋆), "fair" (⋆) and "bad" (○). "Curse of dimensionality" refers to the "burden" of required increase in training statistics and processing time when adding more input variables. See also comments in text. The FDA classifier is not represented here since its properties depend on the chosen function.

Performance comparison

Left figure shows the distribution of signal and background used for test.
Right figure shows the resulting separation using various MVA methods.

ROC curves:

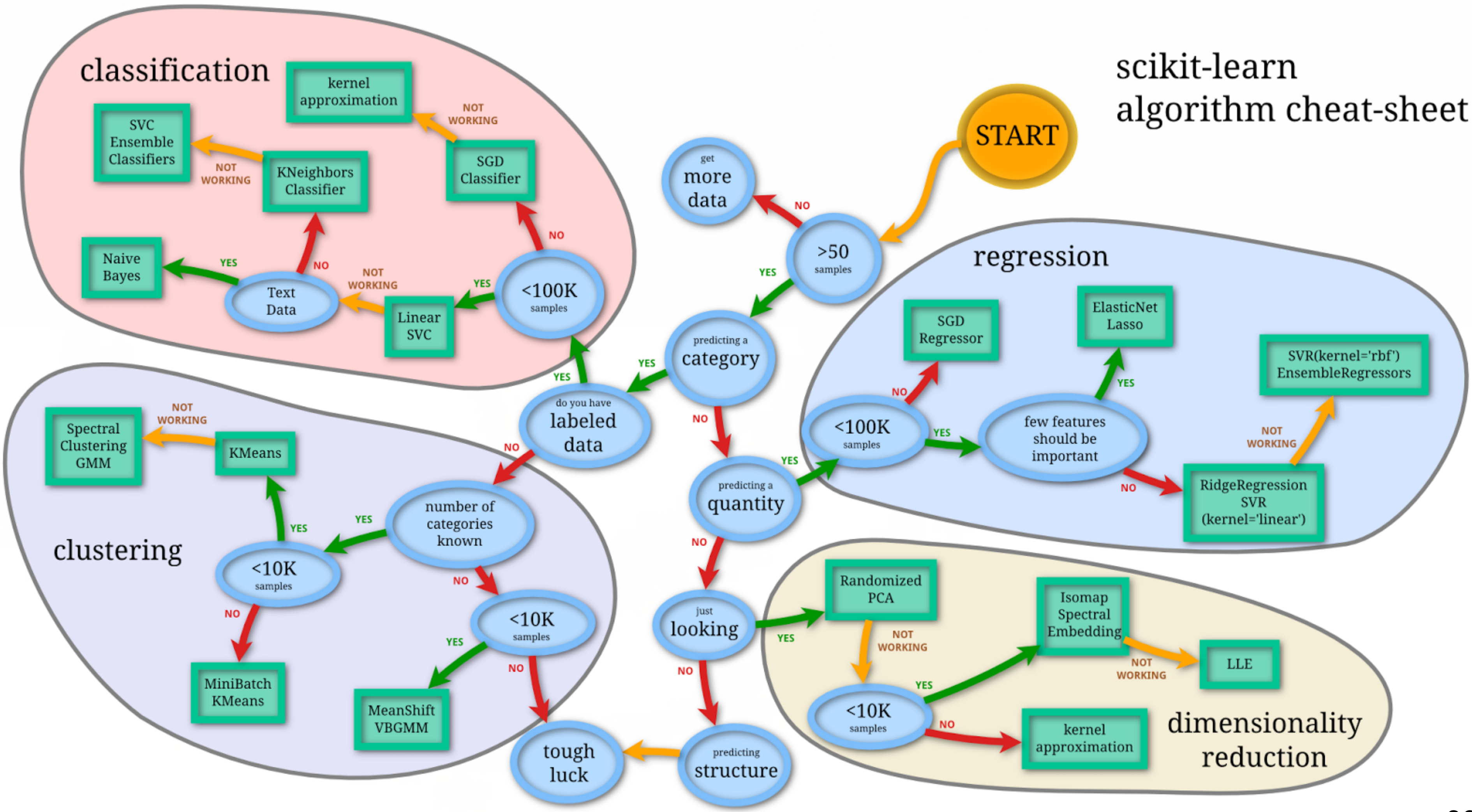




The theoretical limit is known from the Neyman-Pearson lemma using the (known/correct) PDFs in a likelihood.
In all fairness, this is a case that is great for the BDT...

Which method to use?

There is no good / simple answer to this, though people have tried, e.g.:



Which method to use?

There is no good / simple answer to this, though people have tried, e.g.: