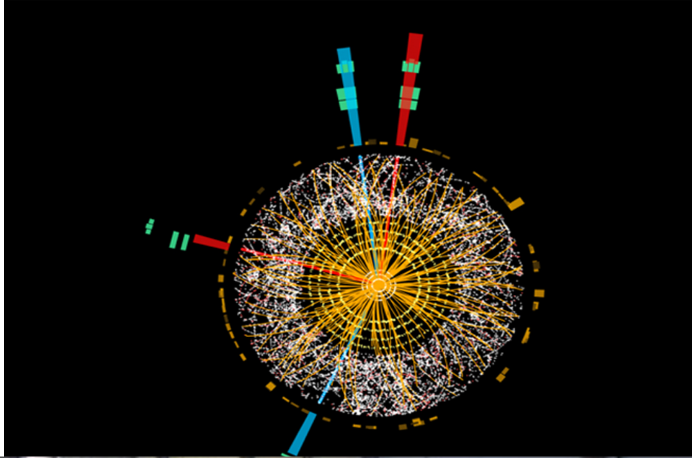
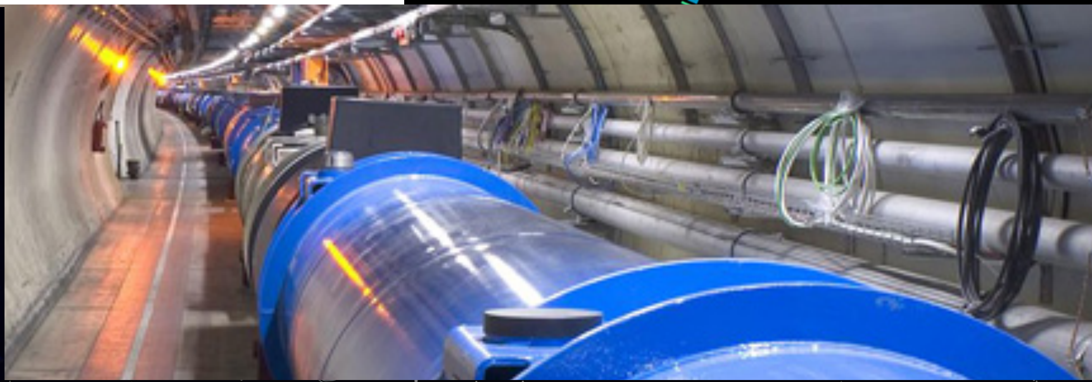
Learning from Data at the Large Hadron Collider

Braga, 26 March 2019





Glen Cowan
Physics Dept.
Royal Holloway, U. London

ROYAL HOLLOWAY UNIVERSITY OF LONDON

Outline

Particle physics: the short story

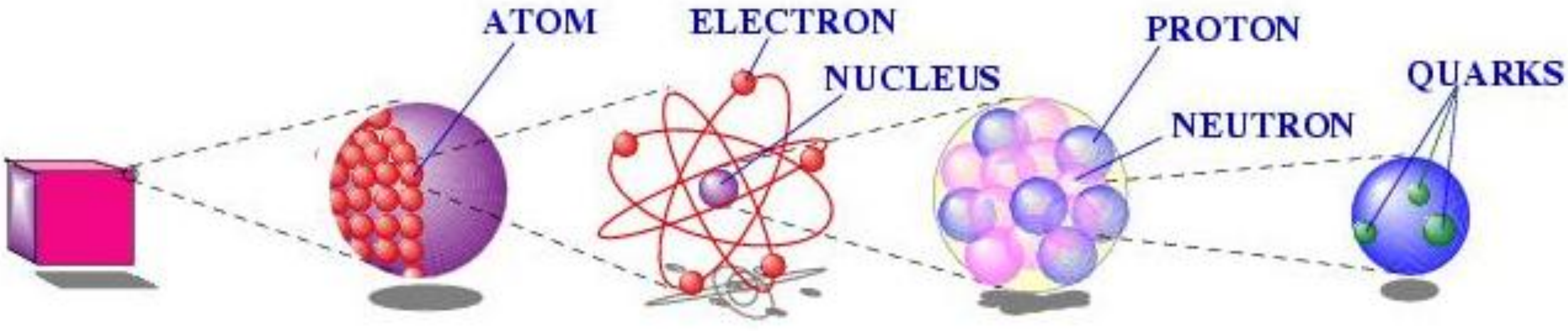
The Large Hadron Collider and the ATLAS Detector

Learning from data: using Machine Learning to

- classify events
- search for New Physics

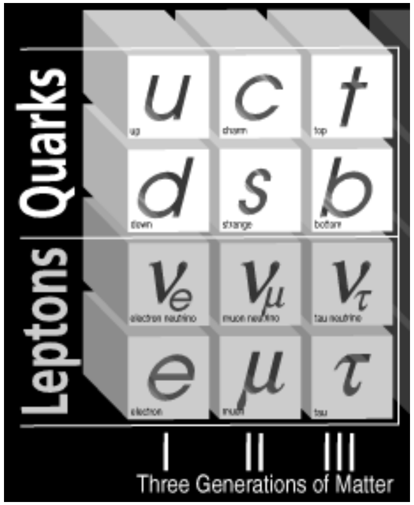
Outlook for Machine Learning and Particle Physics

The Particle Scale



The Current Picture of Particle Physics

Matter...



+ force carriers...

photon (γ)
$W^{\pm}$
Z
gluon (g)

+ Higgs boson + relativity + quantum mechanics + symmetries...

= “The Standard Model”

- almost certainly incomplete
- no gravity yet
- 25 free parameters (!)
- agrees with ~all experimental observations!

Open Questions

What is responsible for dark matter (and dark energy)?

Why is Nature left-right asymmetric? Why three families?

Why does the universe consist almost entirely of matter, rather than a mixture of matter and antimatter?

Why are there 3 space dimensions and 1 time dimension?

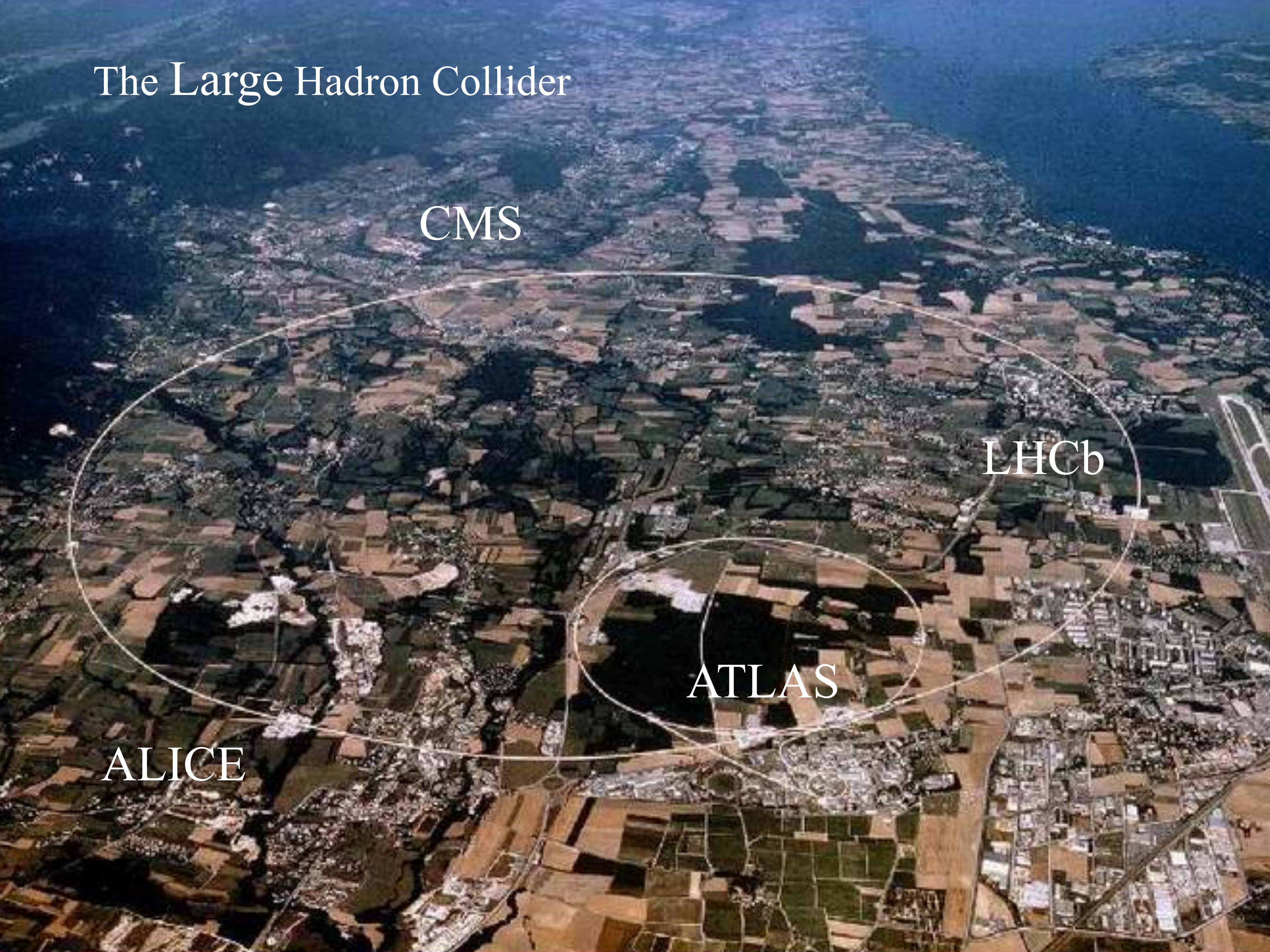
Why is gravity $\sim 10^{40}$ times weaker than electromagnetism?

Why is charge quantised?

Why 25 free parameters?

Theoretical Physicists have proposed a number of alternatives to the Standard Model that address (some of) these questions:

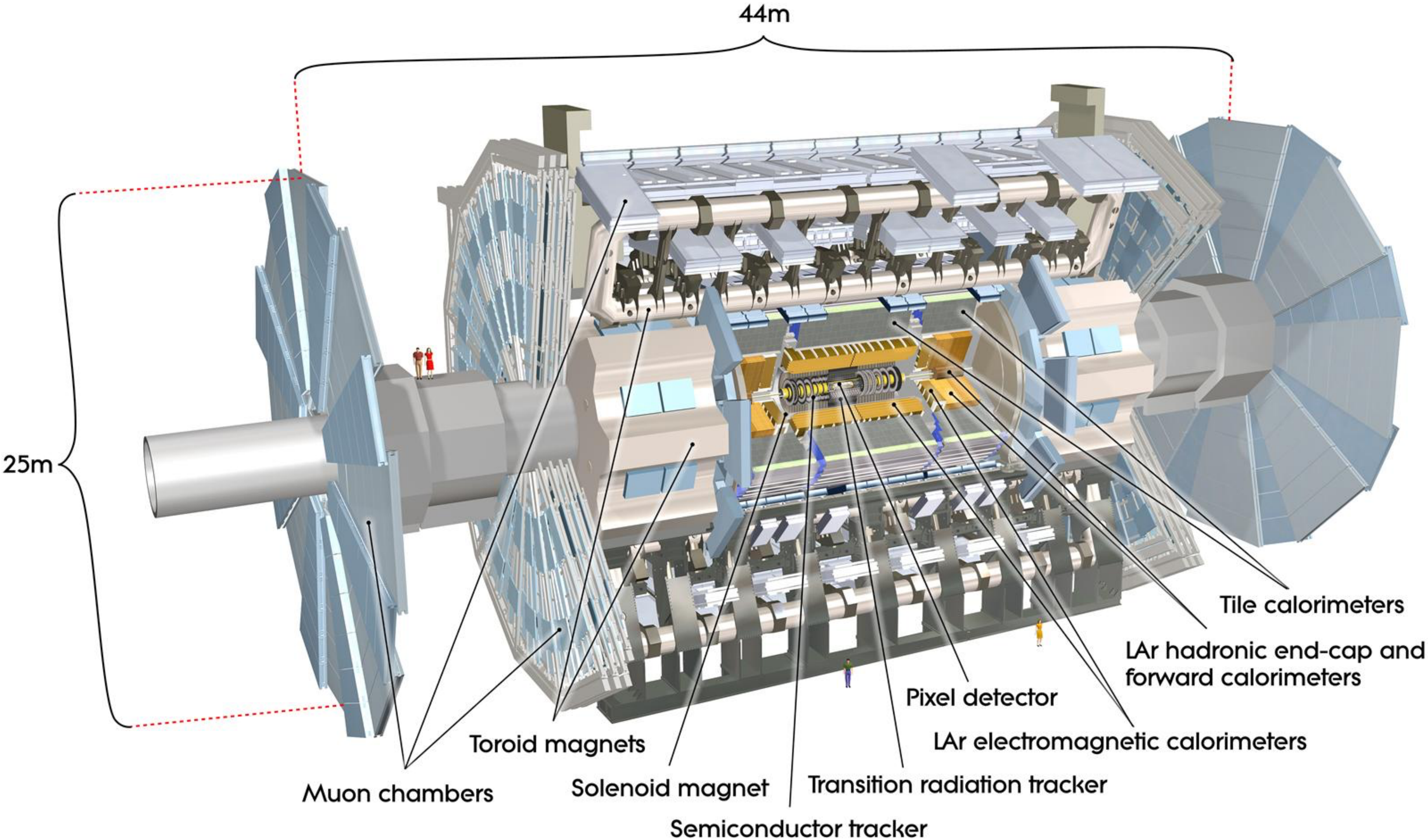
Supersymmetry, Grand Unified Theories, extra dimensions,...


The Large Hadron Collider
CMS
LHCb
ATLAS
ALICE

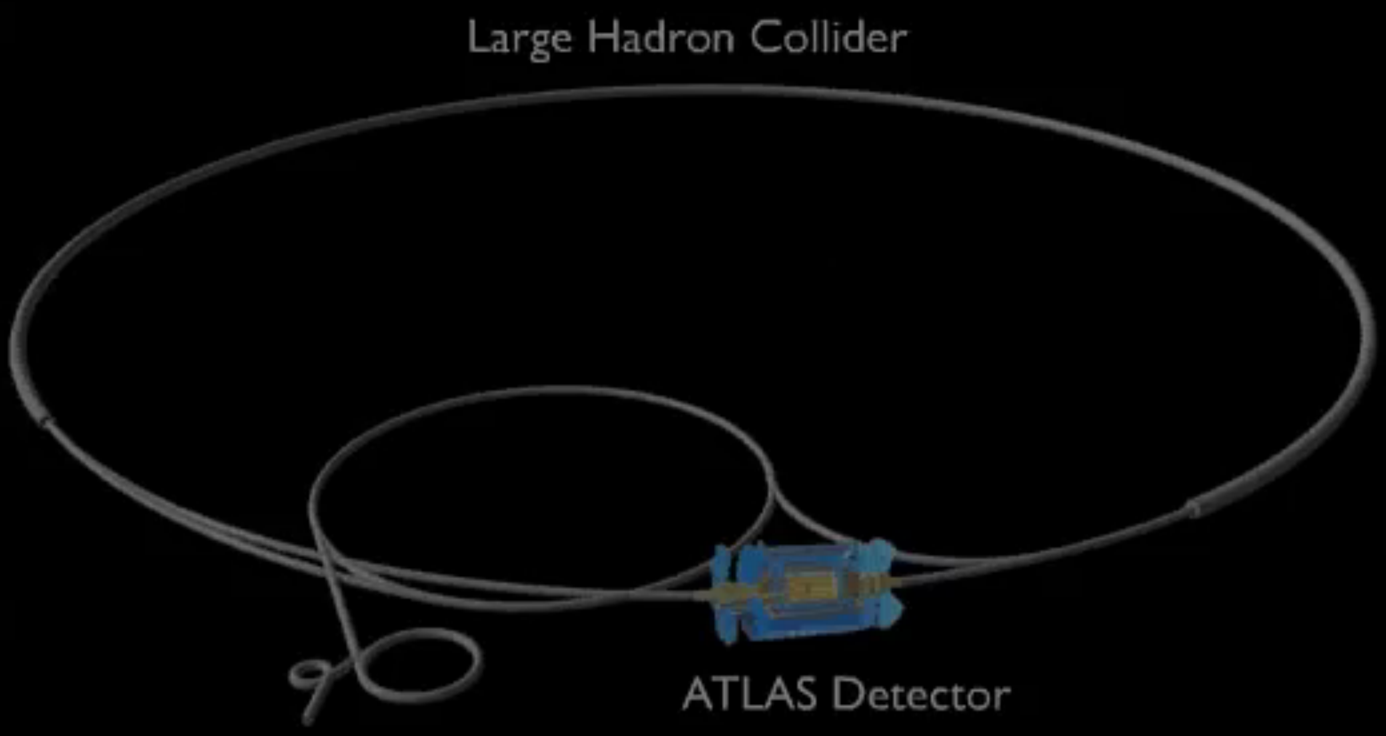

Inside the LHC

The ATLAS Detector at the LHC

The ATLAS Detector



PLAY



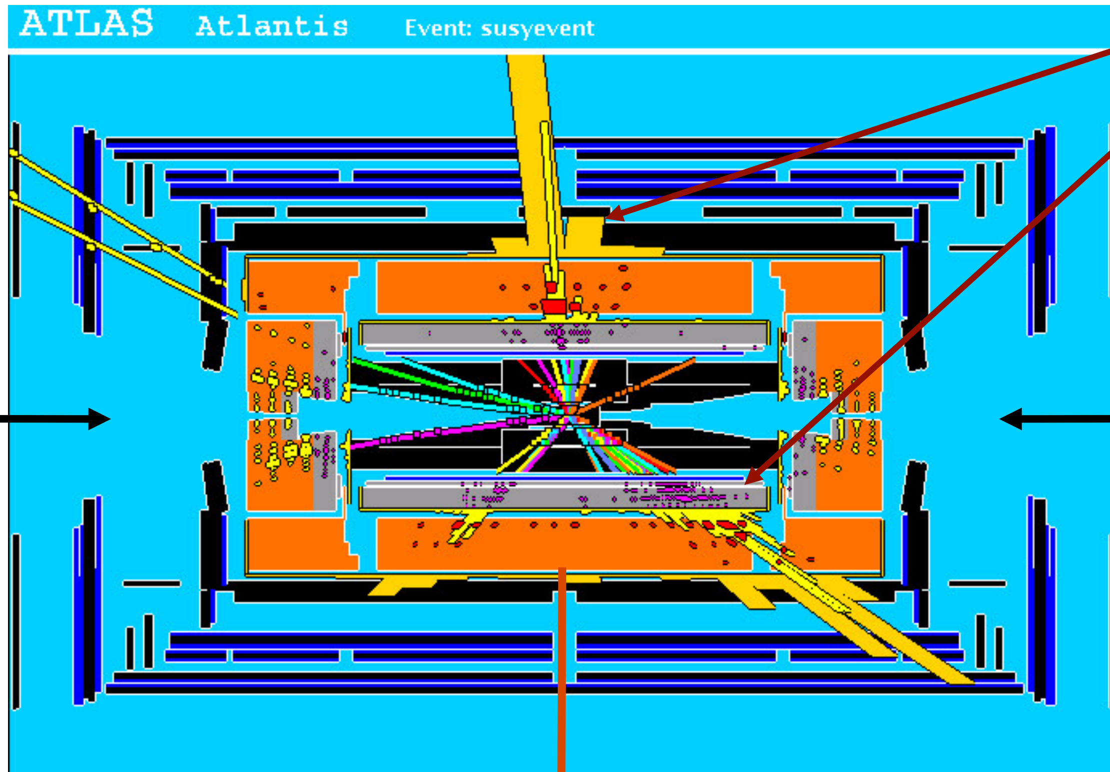
A simulated supersymmetry event

high p_T muons

high p_T jets of hadrons

p

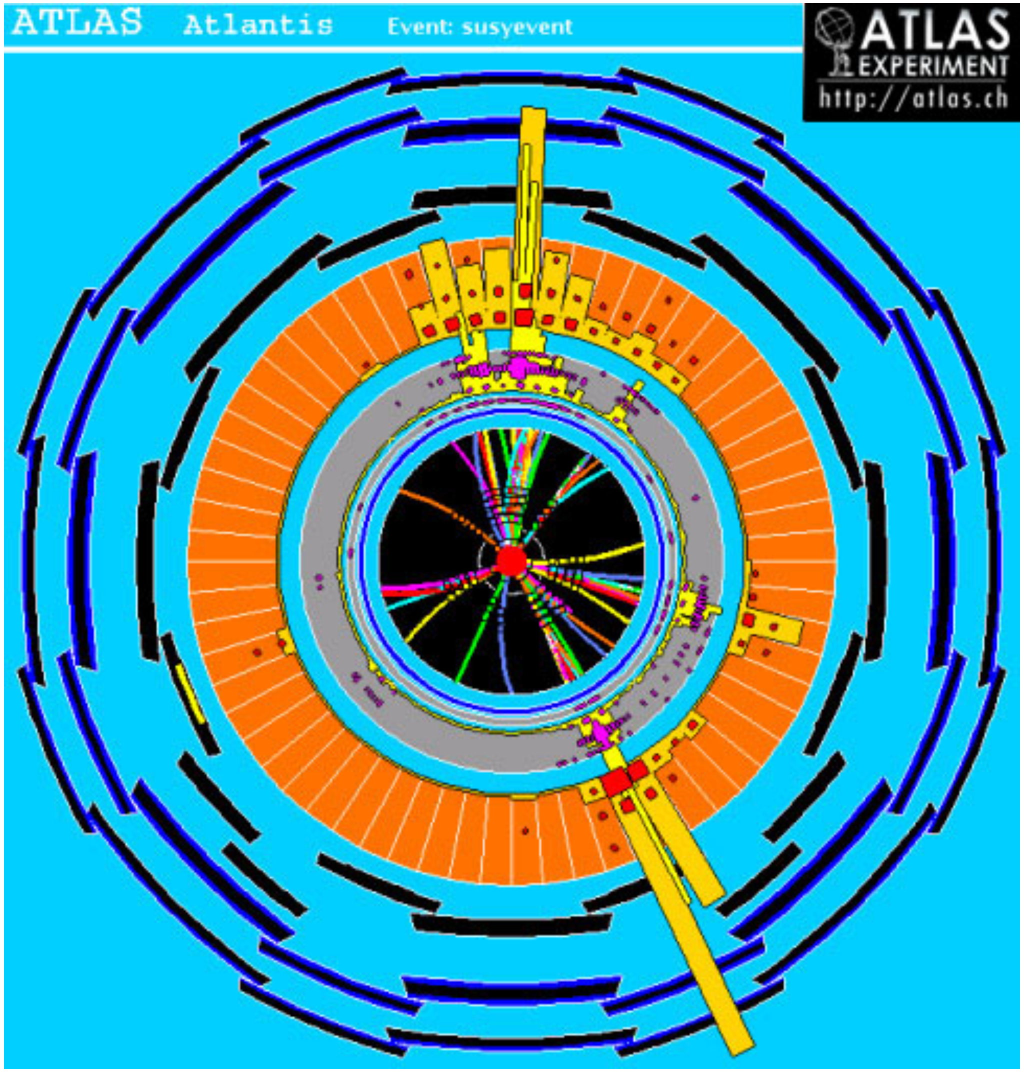
p



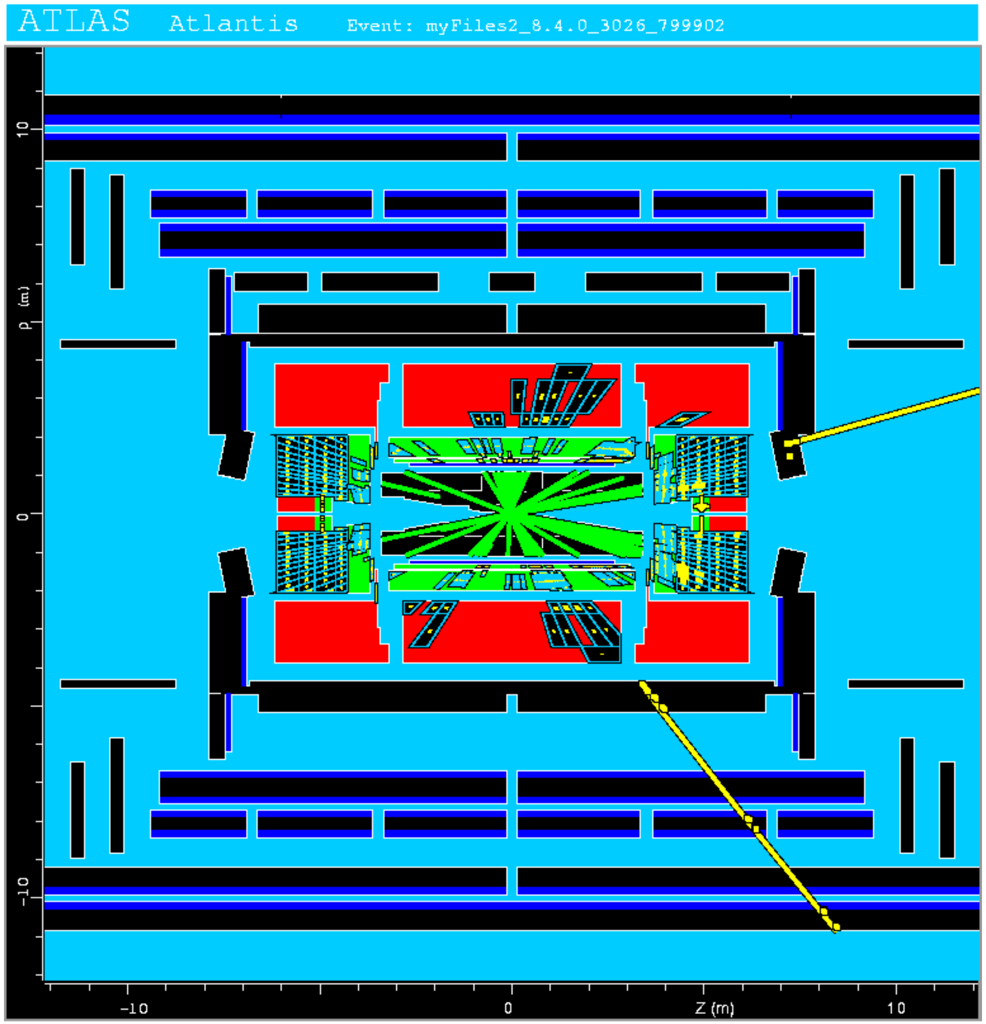
missing energy from undetected neutralinos

Simulated event from supersymmetry

Here momentum imbalance towards upper right indicates that invisible particles (here neutralinos) escaped to lower left ("missing energy").



Background events



This event from Standard Model top-antitop production also has high p_{T} jets and muons, and some missing transverse energy.

→ can easily mimic a SUSY event.

Discovering "New Physics"

For each proton-proton collision, record some set of measured properties (energies and directions of the particles).

Compare the observations to the predictions of different theories and see how well they agree.

If we can find a set of measurements where the data are incompatible with the Standard Model and in good agreement with some alternative, we've made a discovery.

Nobel prize

2^{nd} prize

But the data are "random"! An observed disagreement might be a statistical fluctuation.

Learning from Data

The term Machine Learning (ML) refers to algorithms that "learn from data" and make predictions based on what has been learned.

In its simplest sense, "learning" means the algorithm contains adjustable parameters whose values are estimated using data.

ML can be seen as a part of or related to:

Artificial Intelligence

Pattern Recognition

Statistical Learning

Multivariate Analysis

Development from (mainly) Computer Science, (also) Statistics; sometimes "Data Science" used to refer to all of above.

In Particle Physics, the most important application is the use of classification to search for New Physics.

Example of classification: Industrial Fishing

You scoop up fish which are of two types:

Sea Bass   Cod

You examine the fish with automatic sensors and for each one you measure a set of features:

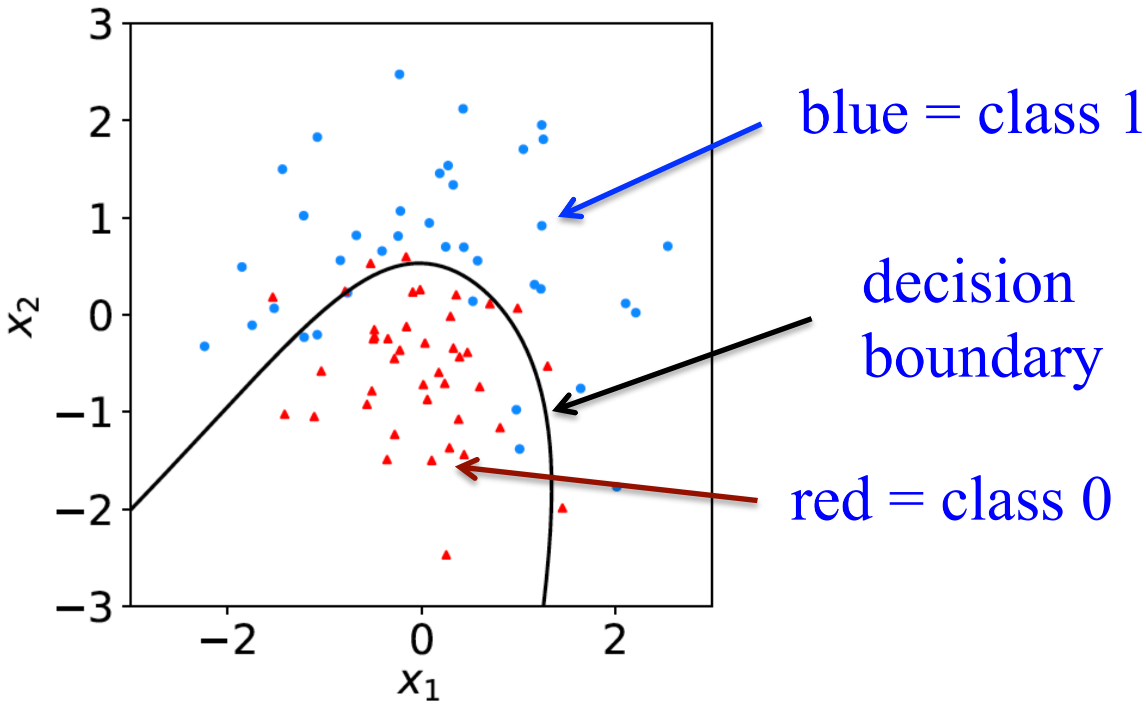
x_1 = length $\quad$ x_4 = area of fins
x_2 = width $\quad$ x_5 = mean spectral reflectance
x_3 = weight $\quad$ x_6 = ...

These constitute the "feature vector" $\boldsymbol{x} = (x_1,\ldots, x_n)$.

In addition you hire a fish expert to identify the "true class label" y = 0 or 1 (i.e., 0 = sea bass, 1 = cod) for each fish.

Distributions of the features

If we consider only two features $\boldsymbol{x} = (x_1, x_2)$, we can display the results in a scatter plot.



Goal is to determine a decision boundary, so that, without the help of the fish expert, we can classify new fish by seeing where their measured features lie relative to the boundary.

Same idea in multi-dimensional feature space, but cannot represent as 2-D plot. Decision boundary is n-dim. hypersurface.

Classification of proton-proton collisions

Proton-proton collisions can be considered to come in two classes:

signal (the kind of event we're looking for, $y = 1$)
background (the kind that mimics signal, $y = 0$)

For each collision (event), we measure a collection of features:

x_1 = energy of muon
x_2 = angle between jets
x_3 = total jet energy
x_4 = missing transverse energy
x_5 = invariant mass of muon pair
x_6 = ...

The real events don't come with true class labels, but computer-simulated events do. So we can have a set of simulated events that consist of a feature vector $\boldsymbol{x}$ and true class label y (0 for background, 1 for signal):

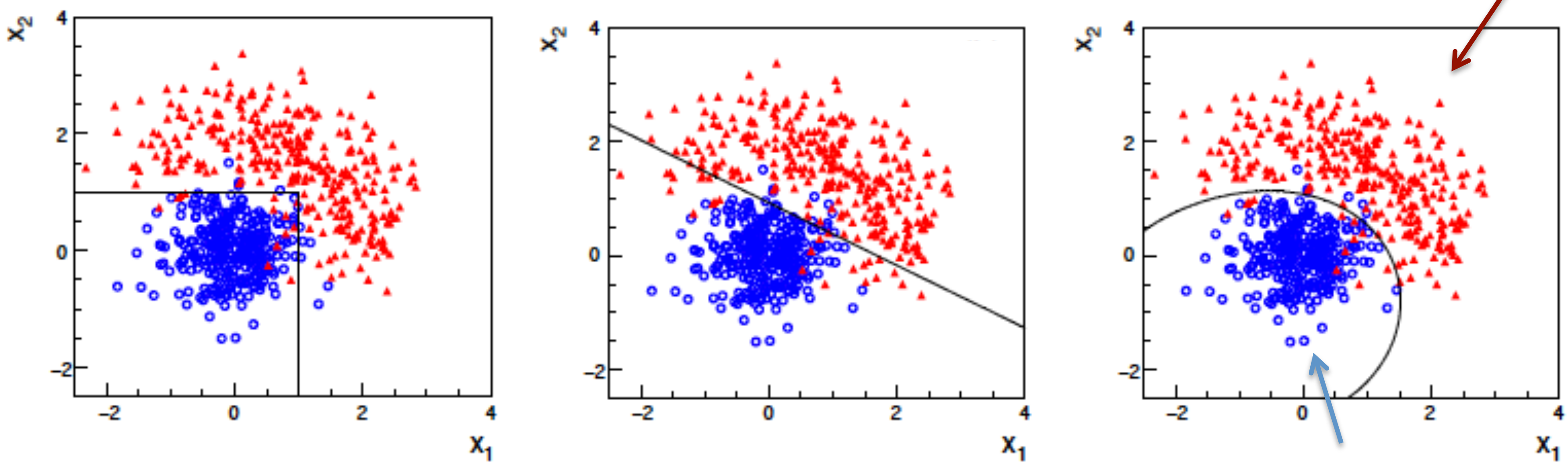
$$(\boldsymbol{x}, y)_1, (\boldsymbol{x}, y)_2, \ldots, (\boldsymbol{x}, y)_N$$

The simulated events are called "training data".

Separating Signal from Background

What is the best “decision boundary”?

signal events



background events

Complications:

The boundary is a hypersurface in a space with, say, tens or hundreds of dimensions.

The events of the signal type may not exist in Nature! Goal is to see if anything “signal-like” is present in the real data.

Mathematics of the decision boundary

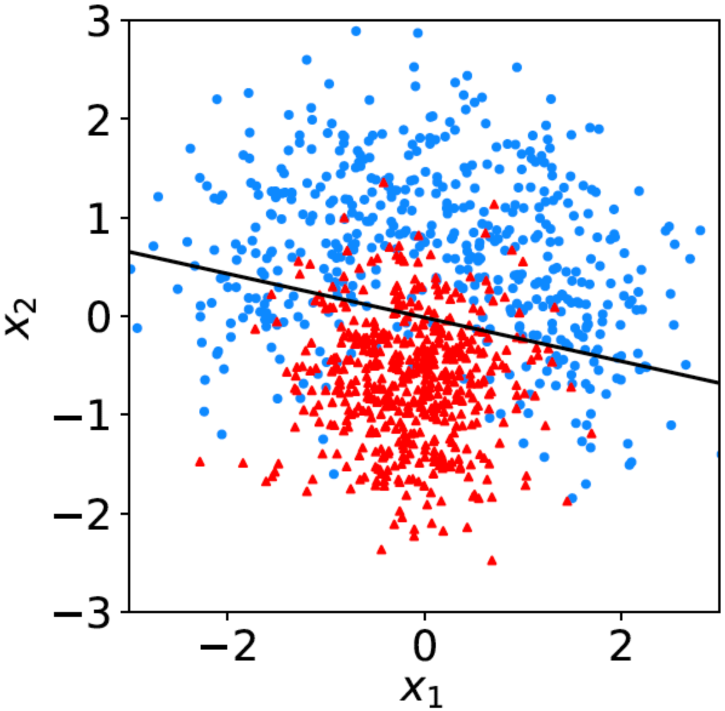
A general surface in the n-dimensional feature space can be described by an equation of the form:

$$t(x_1, \ldots, x_n) = \text{const.}$$

For example, if the function is linear:

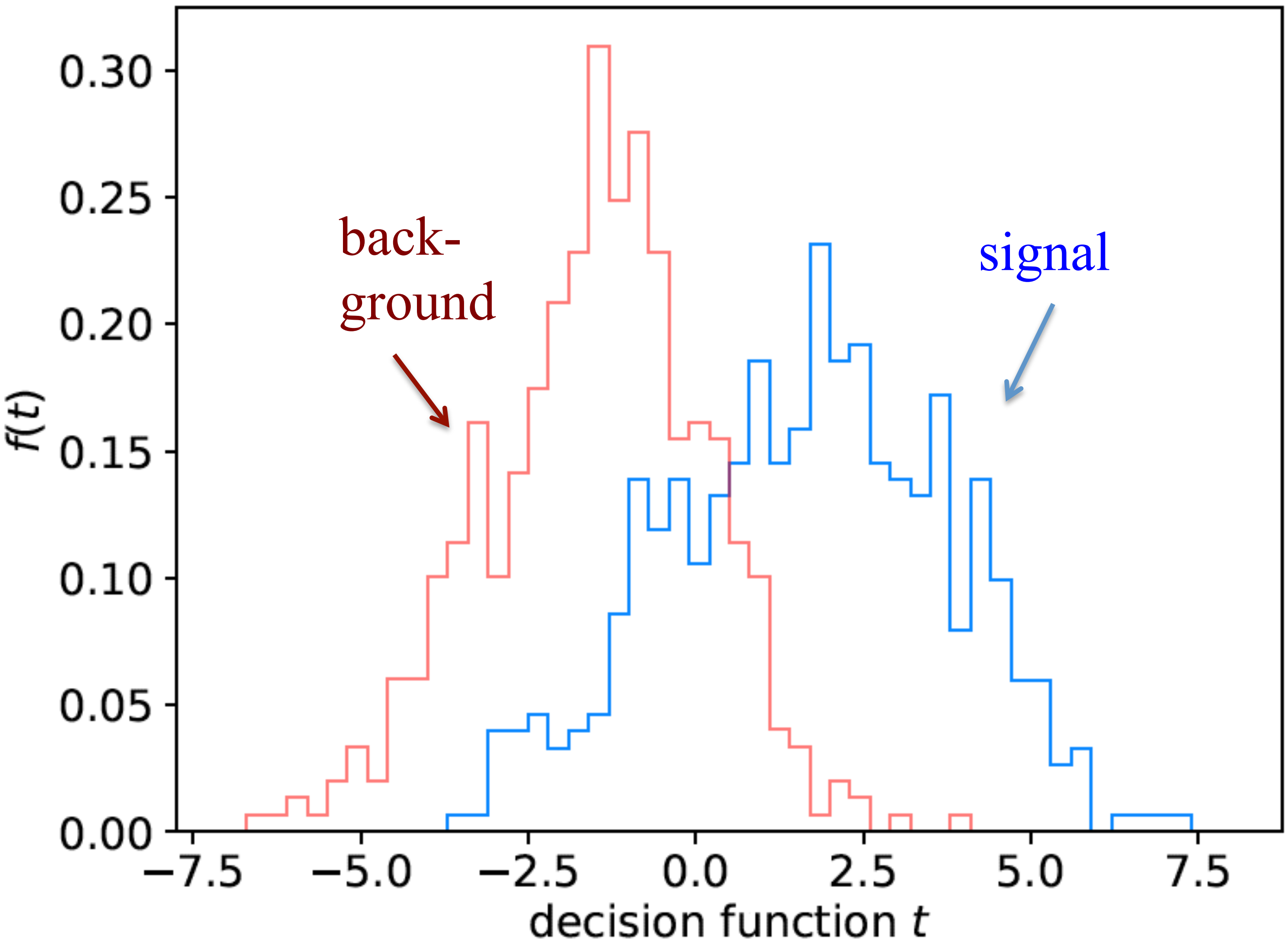
$$c_1 x_1 + c_2 x_2 + \ldots + c_n x_n = \text{const.}$$

then the surface is linear:



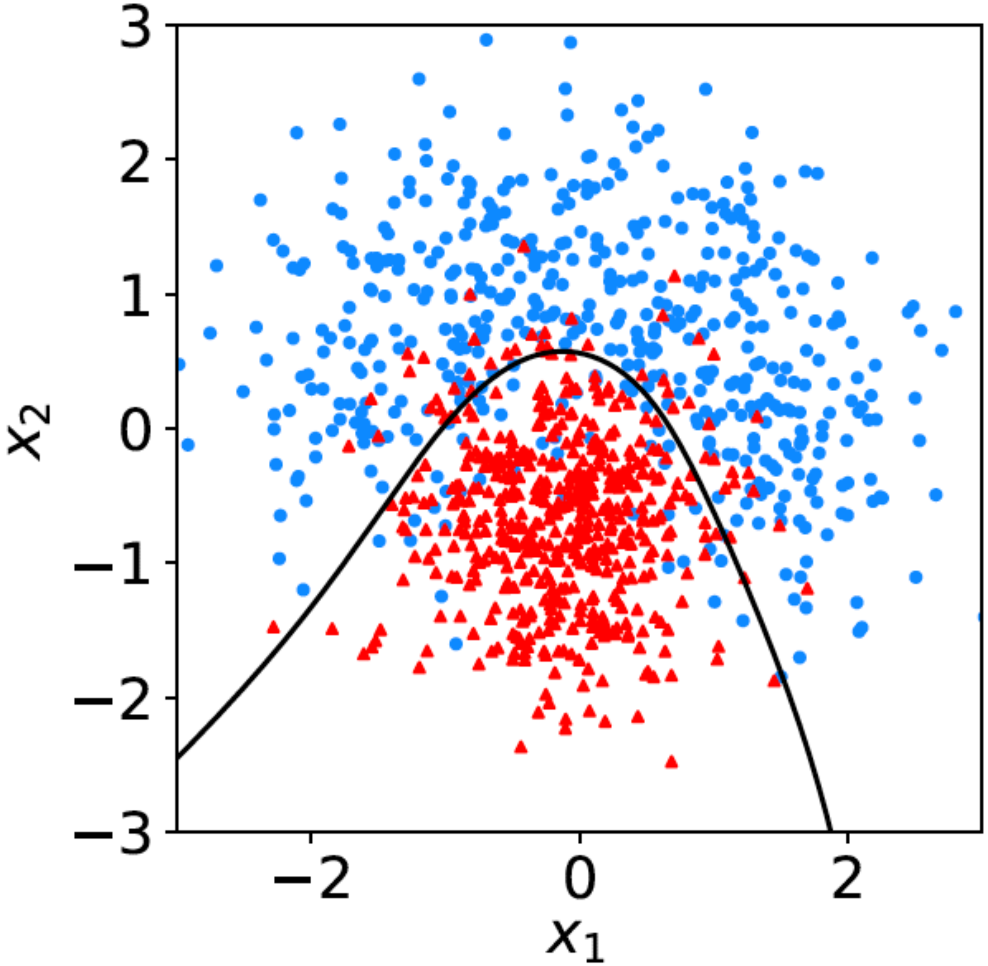
The values of the constants $c_1, c_2, \ldots$ are adjusted using the training data.

Distribution of linear decision function



Nonlinear decision boundaries

From the scatter plot below it's clear that some nonlinear boundary would be better than a linear one:



And to have a nonlinear boundary, the decision function $t(\boldsymbol{x})$ must be nonlinear in $\boldsymbol{x}$.

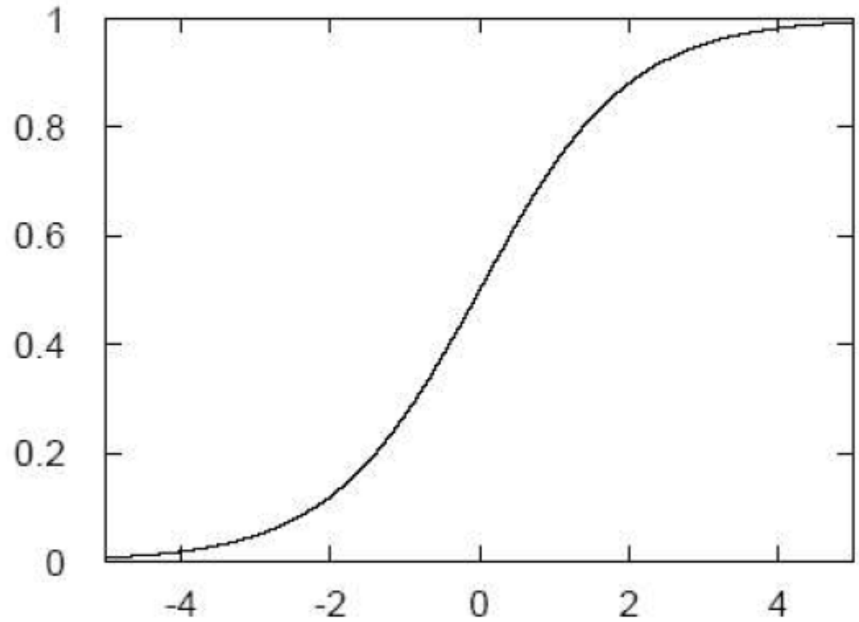
Neural Networks

A simple nonlinear decision function can be constructed as

$$t(\mathbf{x}) = h\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)$$

where h is called the "activation function". For this one can use, e.g., a logistic sigmoid function,

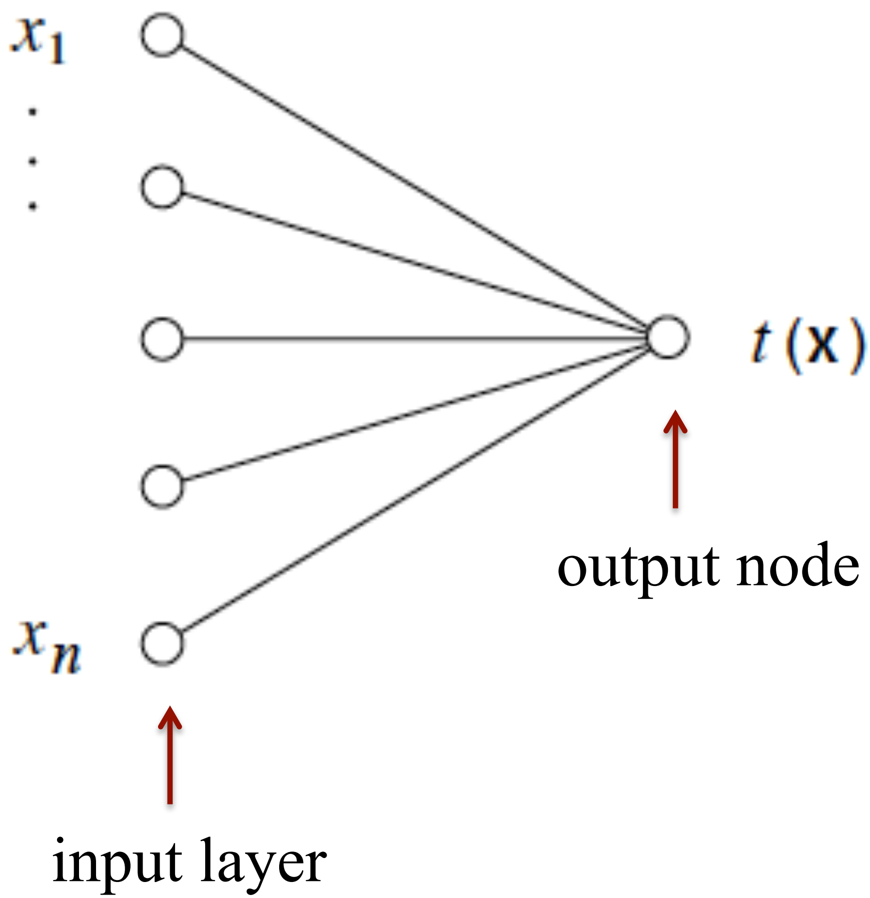
$$h(u) = \frac{1}{1 + e^{-u}}$$



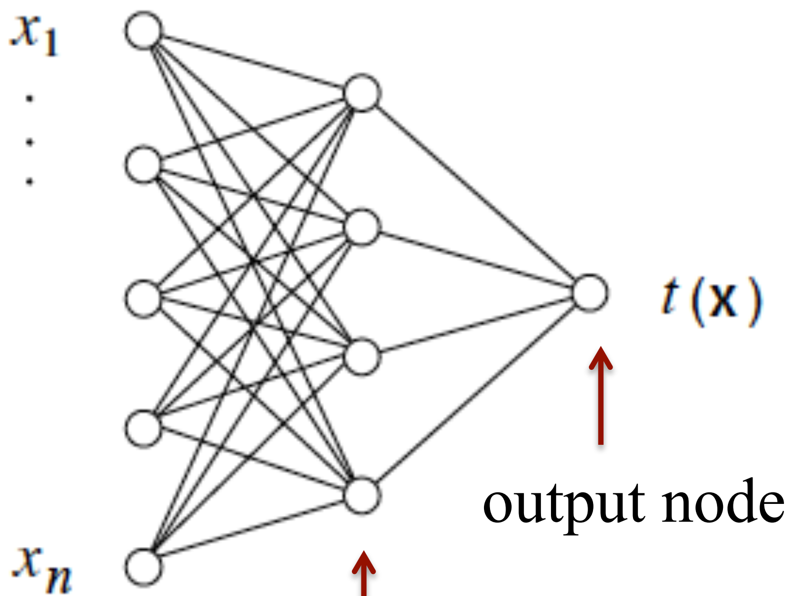
$\longrightarrow u$

Single Layer Perceptron

In this form, the decision function is called a Single Layer Perceptron – the simplest example of a Neural Network.



Multilayer Perceptron



output node

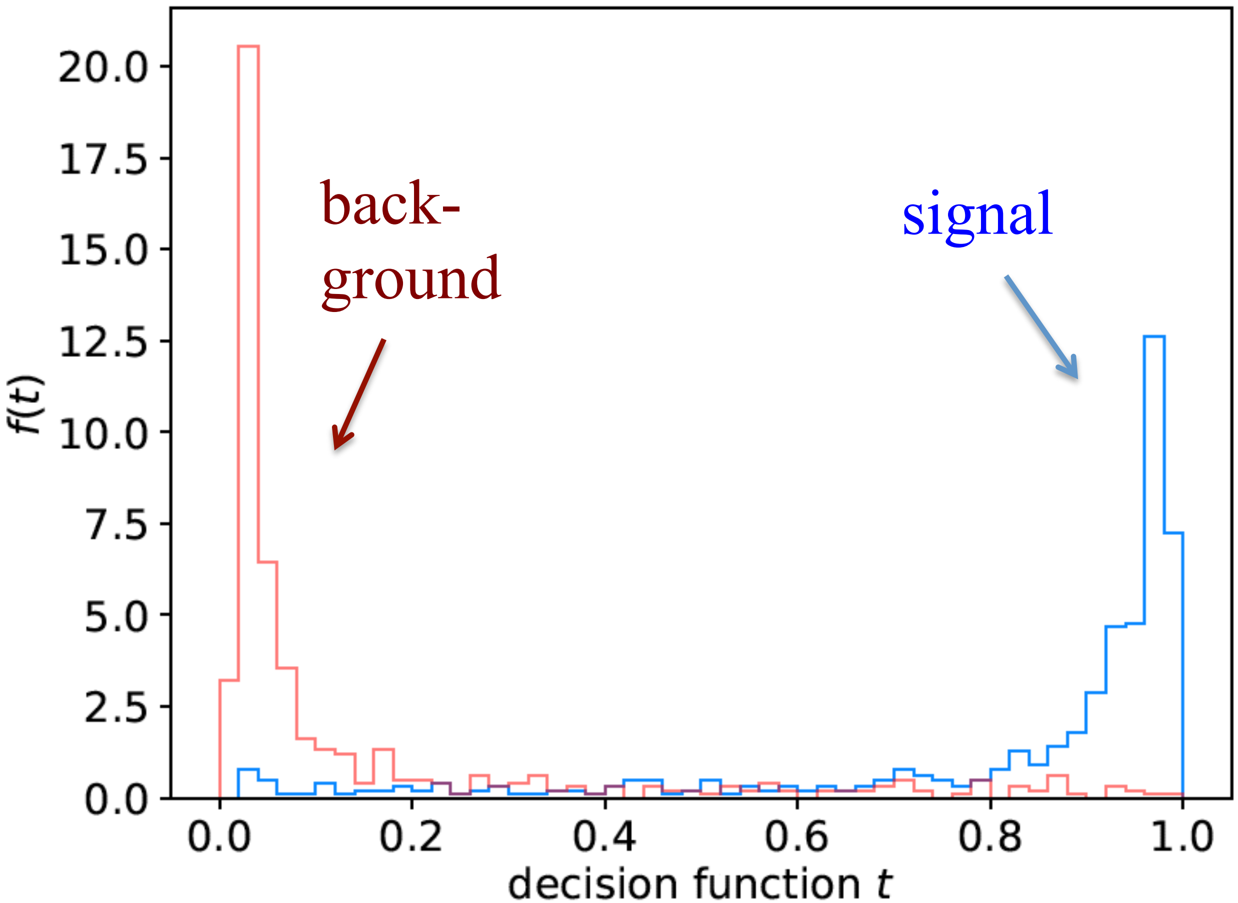
input layer

hidden layer with m nodes $\varphi_1(\boldsymbol{x}),\ldots,\varphi_m(\boldsymbol{x})$

Each line in the graph represents a parameter which must be adjusted using the training data.

Distribution of neural net output

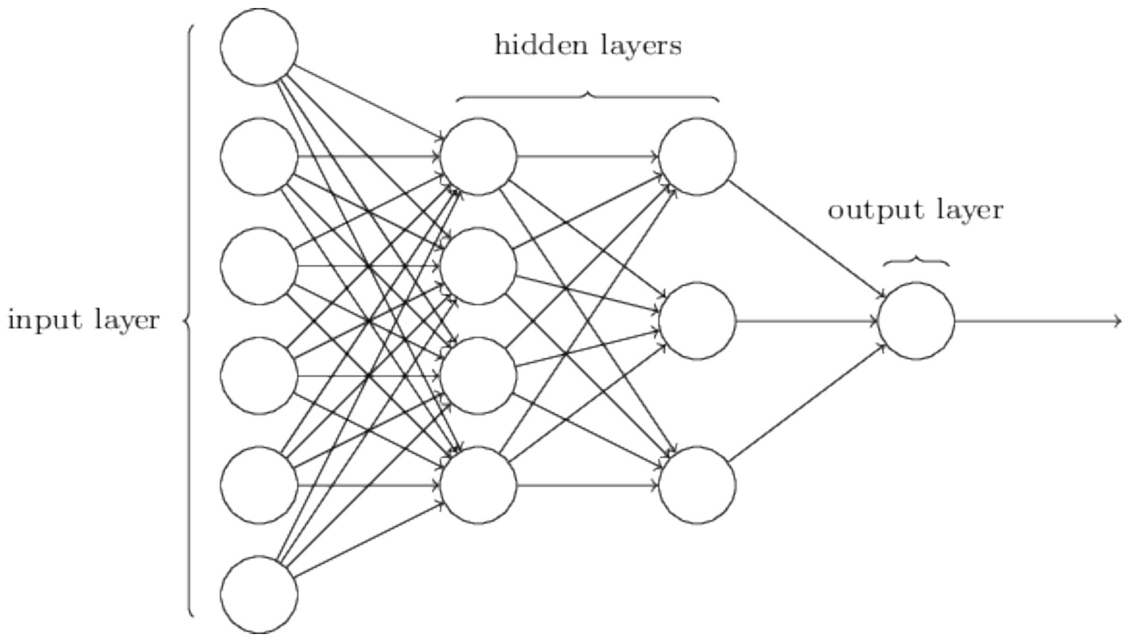
Degree of separation between classes now much better than with linear decision function:



Deep Neural Networks

The multilayer perceptron can have be generalized to have an arbitrary number of hidden layers, with an arbitrary number of nodes in each (= "network architecture").

A "deep" network has several (or many) hidden layers:

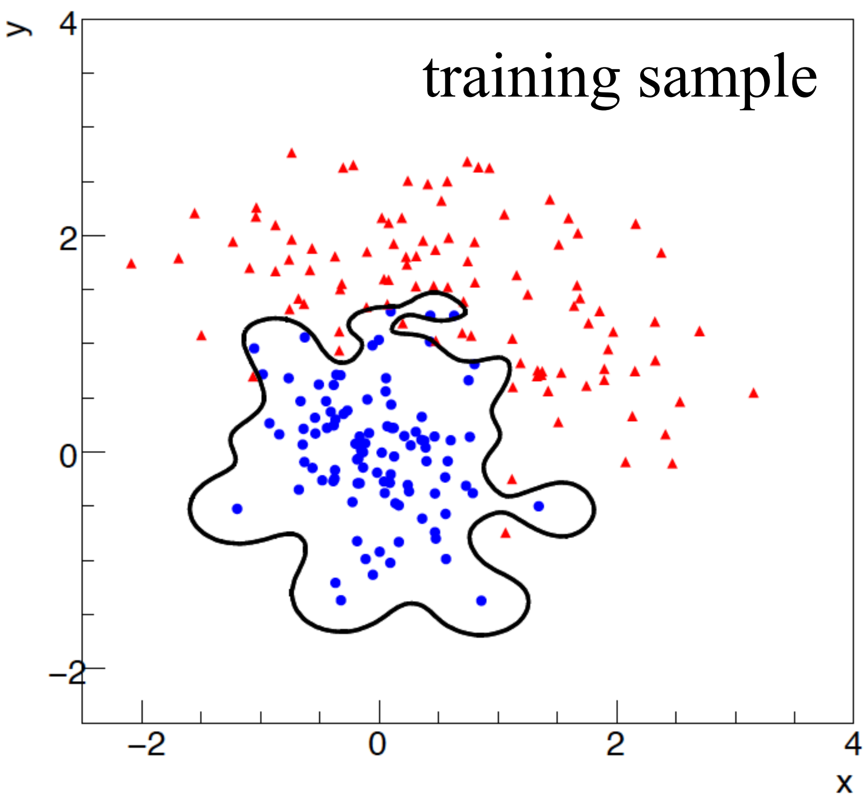


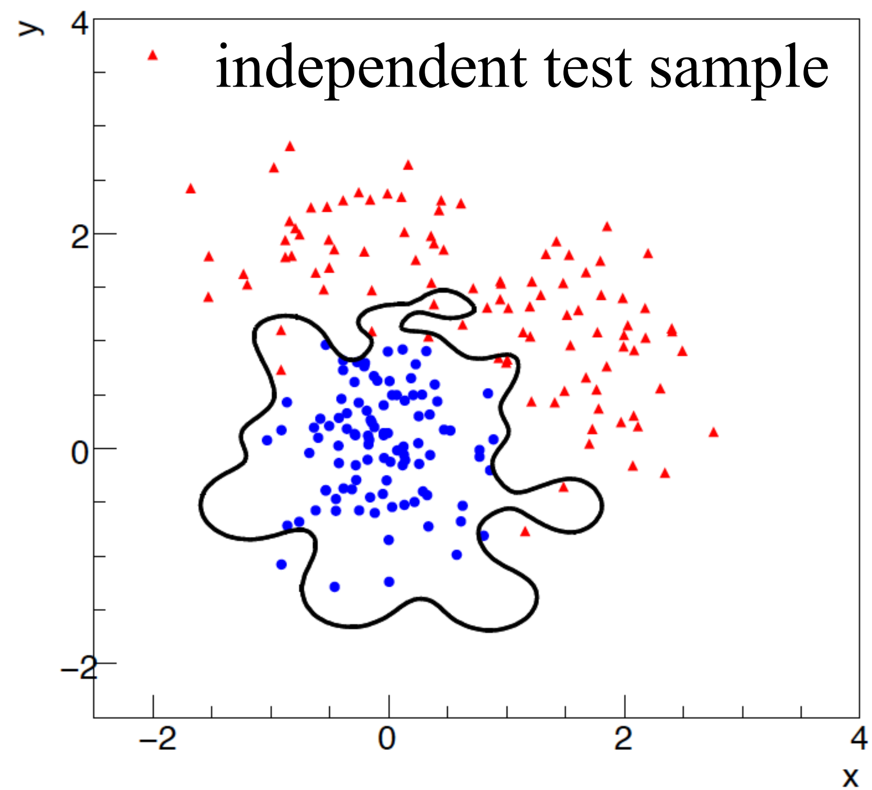
"Deep Learning" is a very recent and active field of research.

Overtraining

Including more adjustable constants in the decision function makes it flexible, and it may conform too closely to the training points.

The same boundary will not perform well on an independent test data sample (→ “overtraining”).





More Machine Learning

Many other ways of defining classifiers:

Support Vector Machines,

Boosted Decision Trees,

K-Nearest Neighbour,

...

There are lots of free software tools, especially with Python:

`scikit-learn.org`

and many online courses; here's a good one (K. Markham):

`www.dataschool.io/machine-learning-with-scikit-learn/`

and here is a website for experimenting with neural networks:

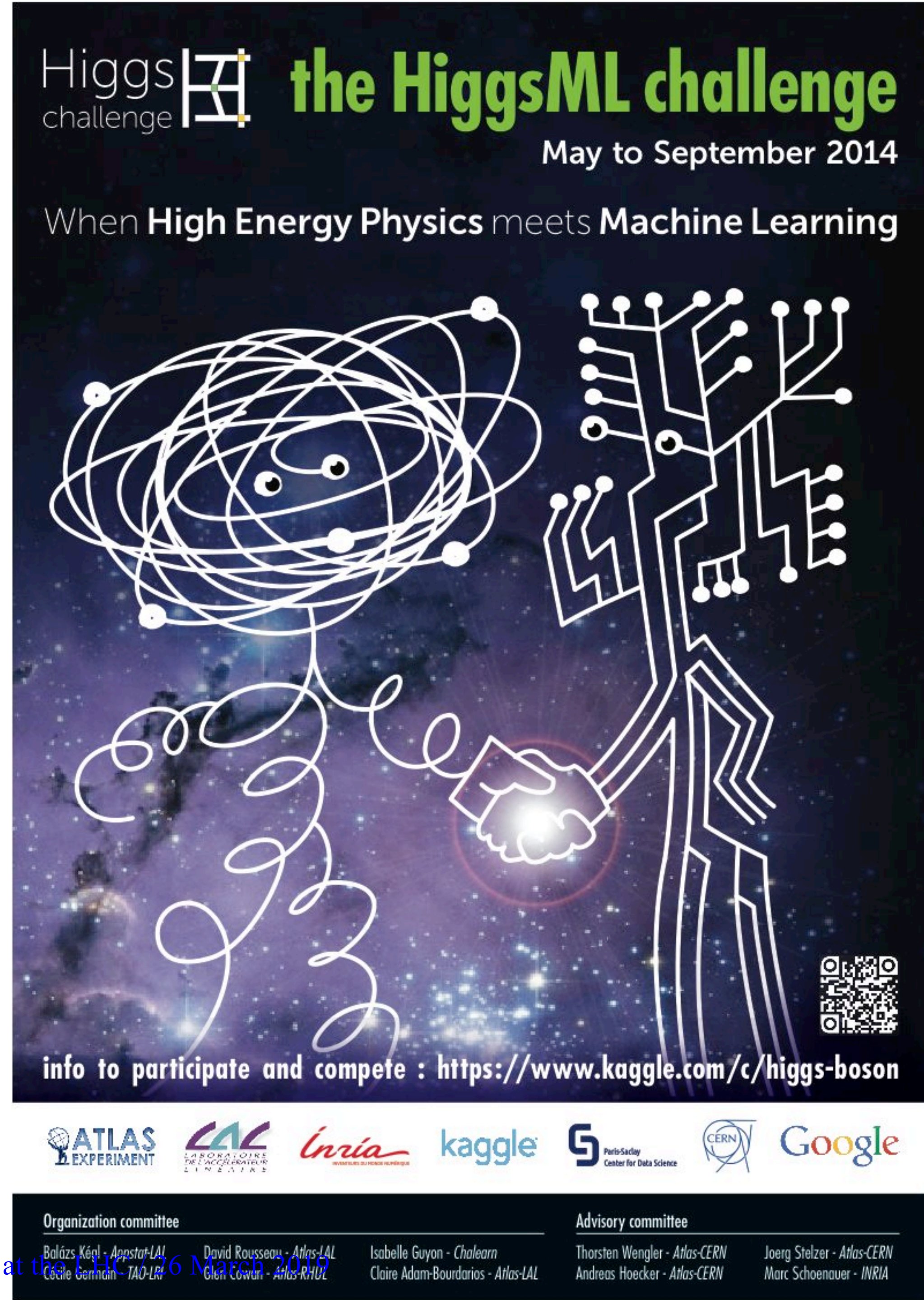
`playground.tensorflow.org`

The Higgs Machine Learning Challenge

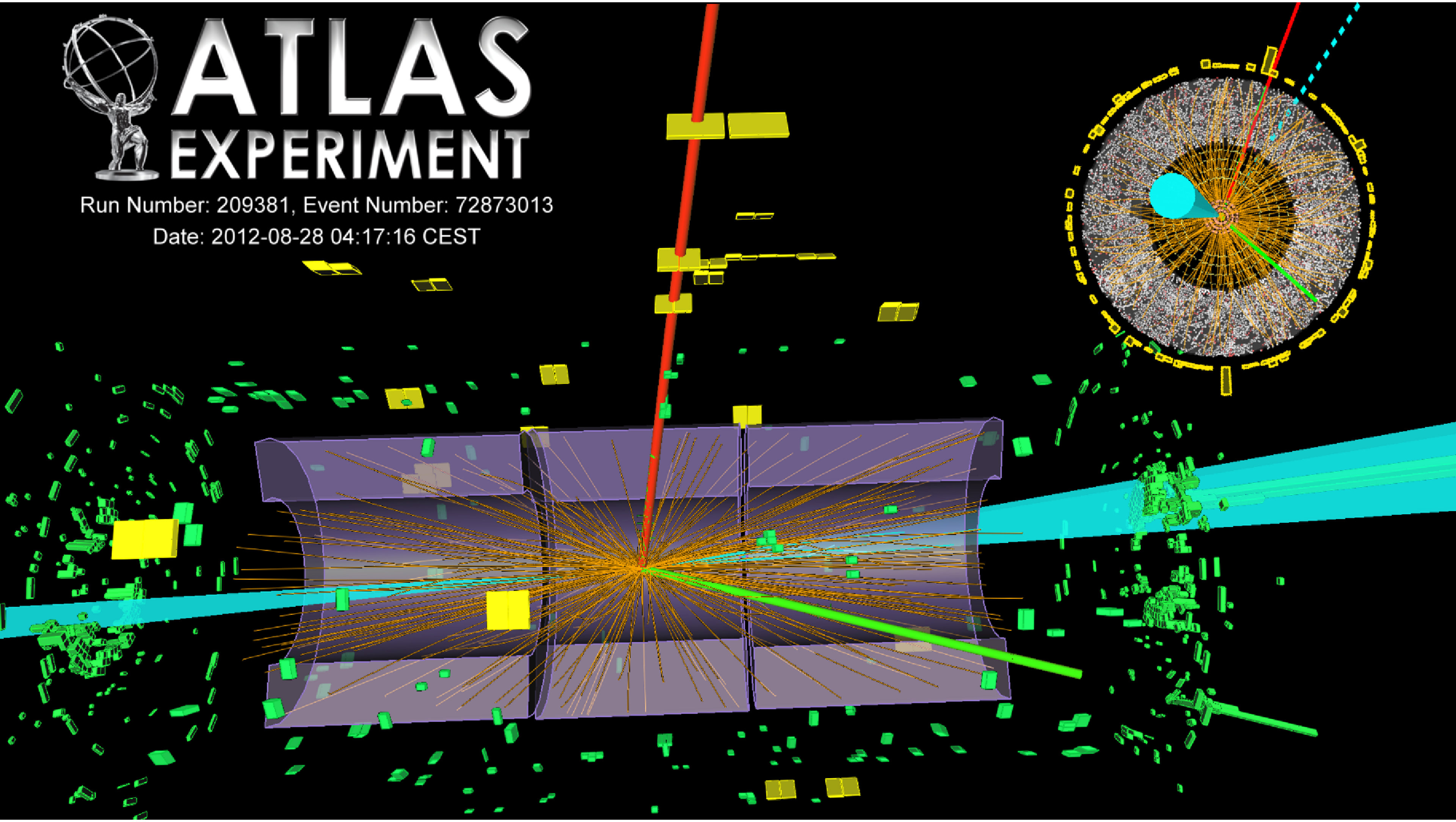
higgsml.lal.in2p3.fr

Highly popular competition to foster exchange of ideas between Machine Learning and Particle Physics

The Challenge: optimise search for Higgs boson decay to pair of tau leptons



Searching for Higgs → tau leptons

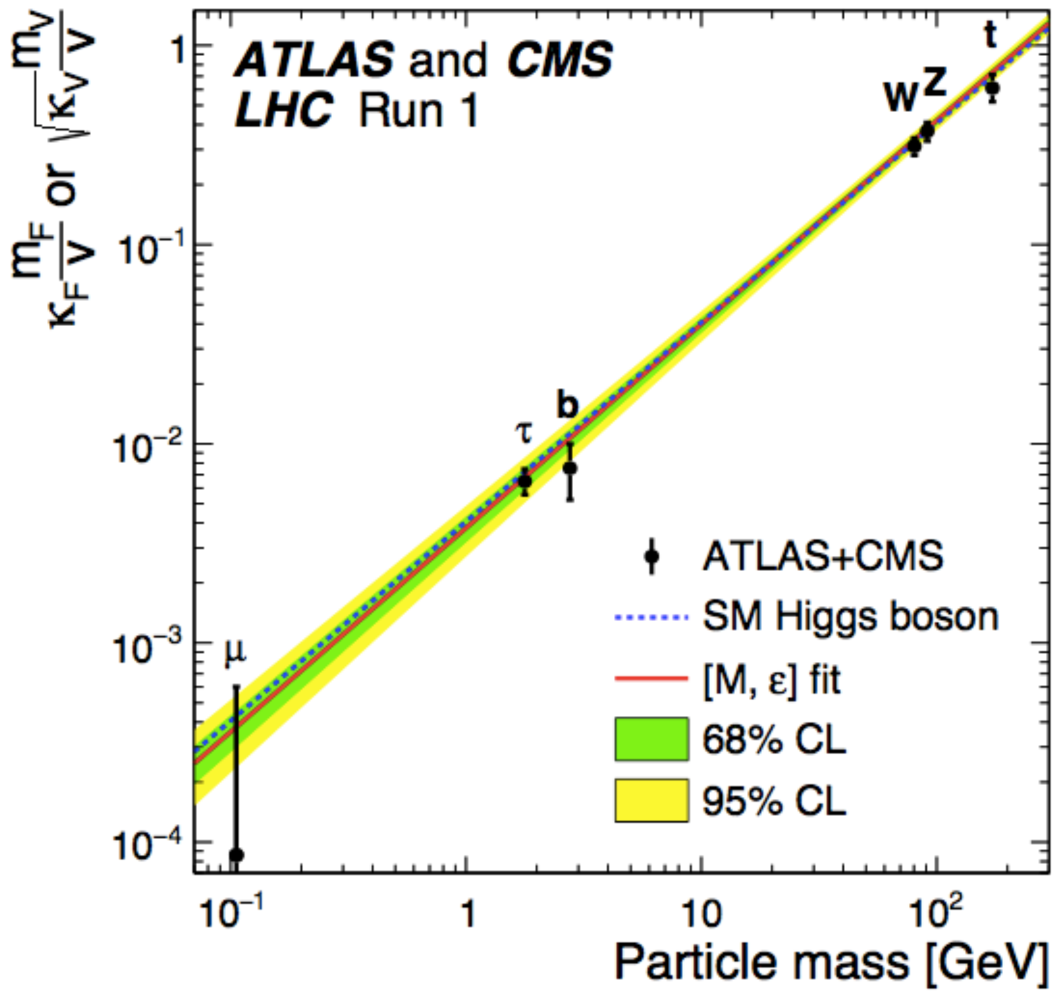


G. Aad et al. [ATLAS and CMS Collaborations], JHEP 1608 (2016) 045 doi:10.1007/JHEP08(2016)045 [arXiv:1606.02266 [hep-ex]].

Coupling strength of Higgs to other particles

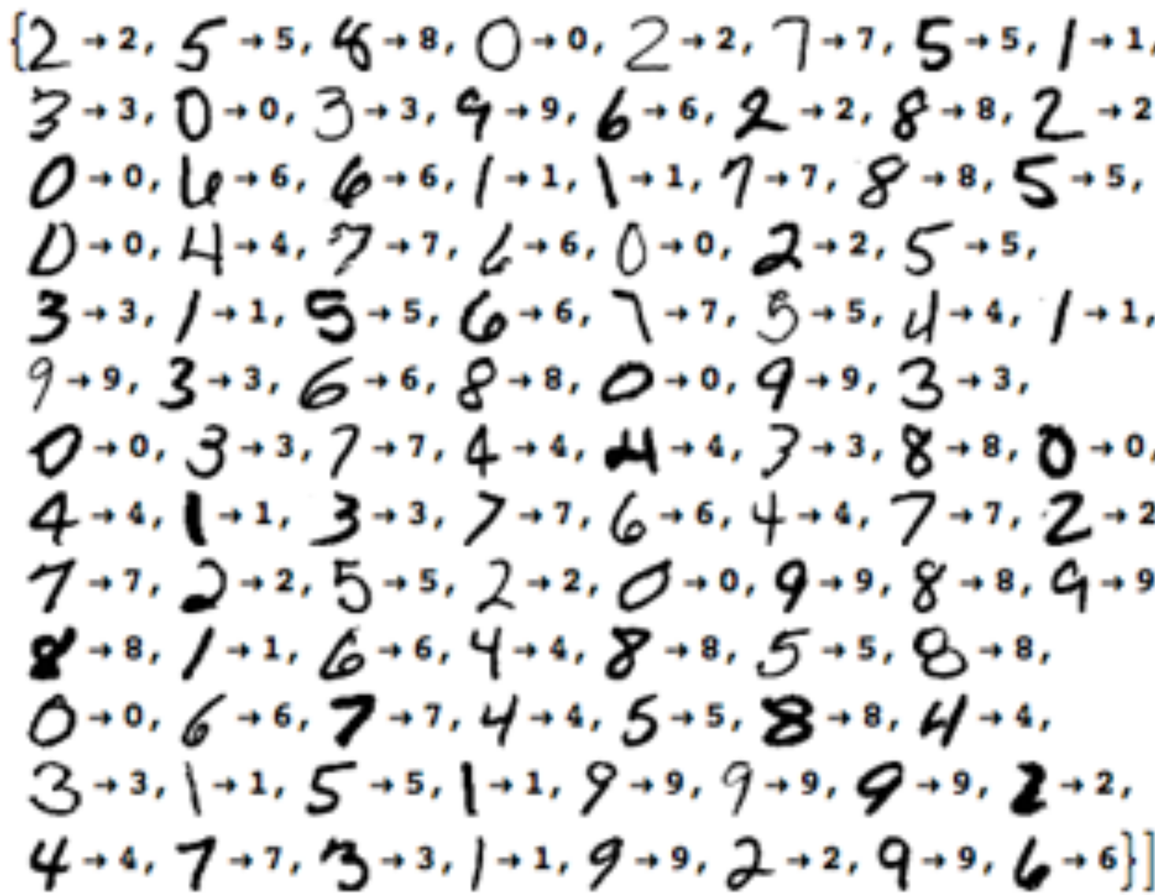
Probability of Higgs decay to a pair of particles of a given type is proportional to the square of the "coupling strength".

Excellent agreement with predictions of Standard Model observed.



Machine Learning for handwriting recognition

Initial feature vector = set of pixels of an image



Scene parsing/labeling with Convolutional Neural Nets

[Farabet et al. ICML 2012, PAMI 2013]

Deep Learning and the Future of AI, seminar at CERN by Yann LeCun: **`https://indico.cern.ch/event/510372/`**

Summary and Outlook

We are continuing to learn about the fundamental particles of Nature with the Large Hadron Collider

Precision measurements of Higgs boson properties

Search for supersymmetry

Search for micro black holes, gravitons, W′, Z′ (extra dimensions)

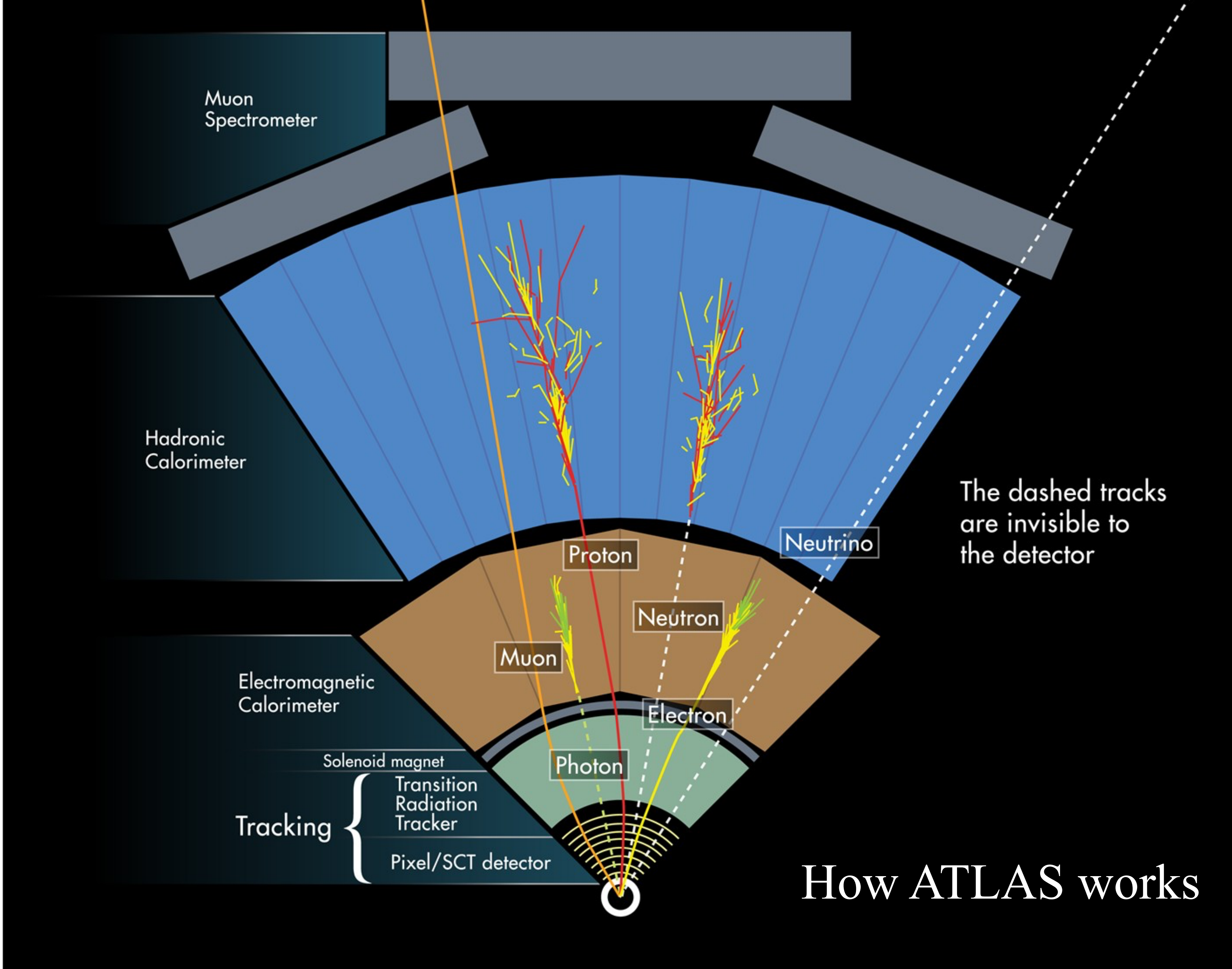
Machine Learning is or soon will be ~everywhere:

Huge interest in Particle Physics, many interdisciplinary initiatives, opportunities for under- and post-grad students.

Lots of accessible software tools (e.g., scikit-learn)

Huge impact on society

Extra slides



How ATLAS works

Simulated "Monte Carlo" Data

Once we define a theory of particle physics, we should in in principle be able to work out the probability for any possible data outcome:

Prob(data | theory)

$$
\begin{aligned}
\mathcal{L} &= \sum_i \overline{\psi}_i \left(i \,\partial\!\!\!/ - m_i - \frac{g m_i H}{2 M_W} \right) \psi_i \\
&- \frac{g}{2\sqrt{2}} \sum_i \overline{\Psi}_i \gamma^\mu (1 - \gamma^5)(T^+ W_\mu^+ + T^- W_\mu^-) \Psi_i \\
&- e \sum_i q_i \overline{\psi}_i \gamma^\mu \psi_i A_\mu \\
&- \frac{g}{2 \cos\theta_W} \sum_i \overline{\psi}_i \gamma^\mu (g_V^i - g_A^i \gamma^5) \psi_i Z_\mu \,.
\end{aligned}
$$

But the calculations are too difficult. Instead we can create computer programs that "generate" simulated data using random numbers (the Monte Carlo method).

We have separate Monte Carlo programs that generate events corresponding to different theories:

Standard Model, Supersymmetry, extra dimensions,...

A generated event

Event listing (summary)

I	particle/jet	KS	KF	orig	p_x	p_y	p_z	E	m
1	!p+!	21	2212	0	0.000	0.000	7000.000	7000.000	0.938
2	!p+!	21	2212	0	0.000	0.000	-7000.000	7000.000	0.938
3	!g!	21	21	1	0.863	-0.323	1739.862	1739.862	0.000
4	!ubar!	21	-2	2	-0.621	-0.163	-777.415	777.415	0.000
5	!g!	21	21	3	-2.427	5.486	1487.857	1487.	
6	!g!	21	21	4	-62.910	63.357	-463.274	471.	
7	!~g!	21	1000021	0	314.363	544.843	498.897	979.	
8	!~g!	21	1000021	0	-379.700	-476.000	525.686	980.	
9	!~chi_1-!	21	-1000024	7	130.058	112.247	129.860	263.	
10	!sbar!	21	-3	7	259.400	187.468	83.100	330.	
11	!c!	21	4	7	-79.403	242.409	283.026	381.	
12	!~chi_20!	21	1000023	8	-326.241	-80.971	113.712	385.	
13	!b!	21	5	8	-51.841	-294.077	389.853	491.	
14	!bbar!	21	-5	8	-0.597	-99.577	21.299	101.	
15	!~chi_10!	21	1000022	9	103.352	81.316	83.457	175.	
16	!s!	21	3	9	5.451	38.374	52.302	65.	
17	!cbar!	21	-4	9	20.839	-7.250	-5.938	22.	
18	!~chi_10!	21	1000022	12	-136.266	-72.961	53.246	181.	
19	!nu_mu!	21	14	12	-78.263	-24.757	21.719	84.	
20	!nu_mubar!	21	-14	12	-107.801	16.901	38.226	115.	
21	gamma	1	22	4	2.636	1.357	0.125	2.	
22	(~chi_1-)	11	-1000024	9	129.643	112.440	129.820	262.	
23	(~chi_20)	11	1000023	12	-322.330	-80.817	113.191	382.	
24	~chi_10	1	1000022	15	97.944	77.819	80.917	169.	
25	~chi_10	1	1000022	18	-136.266	-72.961	53.246	181.	
26	nu_mu	1	14	19	-78.263	-24.757	21.719	84.	
27	nu_mubar	1	-14	20	-107.801	16.901	38.226	115.	
28	(Delta++)	11	2224	2	0.222	0.012	-2734.287	2734.	

397	pi+	1	211	209	0.006	0.398	-308.296	308.297	0.140
398	gamma	1	22	211	0.407	0.087	-1695.458	1695.458	0.000
399	gamma	1	22	211	0.113	-0.029	-314.822	314.822	0.000
400	(pi0)	11	111	212	0.021	0.122	-103.709	103.709	0.135
401	(pi0)	11	111	212	0.084	-0.068	-94.276	94.276	0.135
402	(pi0)	11	111	212	0.267	-0.052	-144.673	144.674	0.135
403	gamma	1	22	215	-1.581	2.473	3.306	4.421	0.000
404	gamma	1	22	215	-1.494	2.143	3.051	4.016	0.000
405	pi-	1	-211	216	0.007	0.738	4.015	4.085	0.140
406	pi+	1	211	216	-0.024	0.293	0.486	0.585	0.140
407	K+	1	321	218	4.382	-1.412	-1.799	4.968	0.494
408	pi-	1	-211	218	1.183	-0.894	-0.176	1.500	0.140
409	(pi0)	11	111	218	0.955	-0.459	-0.590	1.221	0.135
410	(pi0)	11	111	218	2.349	-1.105	-1.181	2.855	0.135
411	(Kbar0)	11	-311	219	1.441	-0.247	-0.472	1.615	0.498
412	pi-	1	-211	219	2.232	-0.400	-0.249	2.285	0.140
413	K+	1	321	220	1.380	-0.652	-0.361	1.644	0.494
414	(pi0)	11	111	220	1.078	-0.265	0.175	1.132	0.135
415	(K_S0)	11	310	222	1.841	0.111	0.894	2.109	0.498
416	K+	1	321	223	0.307	0.107	0.252	0.642	0.494
417	pi-	1	-211	223	0.266	0.316	-0.201	0.480	0.140
418	nbar0	1	-2112	226	1.335	1.641	2.078	3.111	0.940
419	(pi0)	11	111	226	0.899	1.046	1.311	1.908	0.135
420	pi+	1	211	227	0.217	1.407	1.356	1.971	0.140
421	(pi0)	11	111	227	1.207	2.336	2.767	3.820	0.135
422	n0	1	2112	228	3.475	5.324	5.702	8.592	0.940
423	pi-	1	-211	228	1.856	2.606	2.808	4.259	0.140
424	gamma	1	22	229	-0.012	0.247	0.421	0.489	0.000
425	gamma	1	22	229	0.025	0.034	0.009	0.043	0.000
426	pi+	1	211	230	2.718	5.229	6.403	8.703	0.140
427	(pi0)	11	111	230	4.109	6.747	7.597	10.961	0.135
428	pi-	1	-211	231	0.551	1.233	1.945	2.372	0.140
429	(pi0)	11	111	231	0.645	1.141	0.922	1.608	0.135
430	gamma	1	22	232	-0.383	1.169	1.208	1.724	0.000
431	gamma	1	22	232	-0.201	0.070	0.060	0.221	0.000

PYTHIA Monte Carlo

pp → gluino-gluino

Training data

The most widely used Machine Learning algorithms used in Particle Physics involve "supervised learning" – this requires samples of data where the type of event is known.

Nature does not provide labels for the real data, so for this we use the simulated (Monte Carlo) data.

So for two event types (signal and background) we have simulated events each with a feature vector and true class label.

A simple example (2D)

Consider two variables, x_1 and x_2, and suppose we have formulas for the joint pdfs for both signal (s) and background (b) events (in real problems the formulas are usually notavailable).

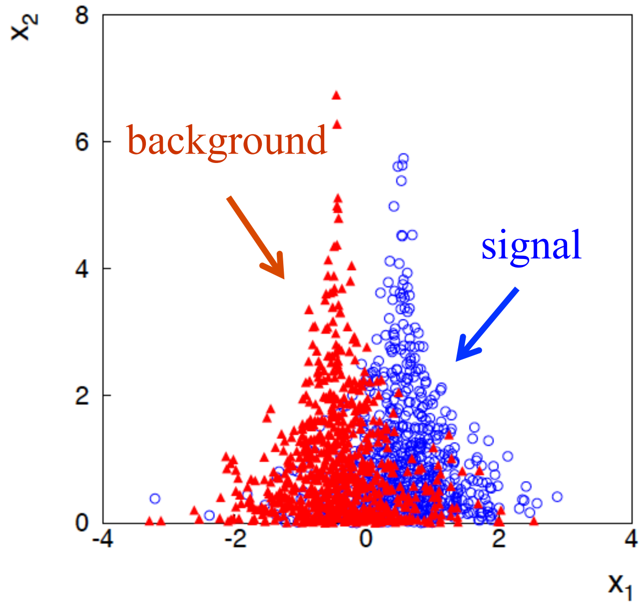
$f(x_1|x_2)$ ~ Gaussian, different means for s/b,
Gaussians have same σ, which depends on x_2,
$f(x_2)$ ~ exponential, same for both s and b,
$f(x_1, x_2) = f(x_1|x_2)\, f(x_2)$:

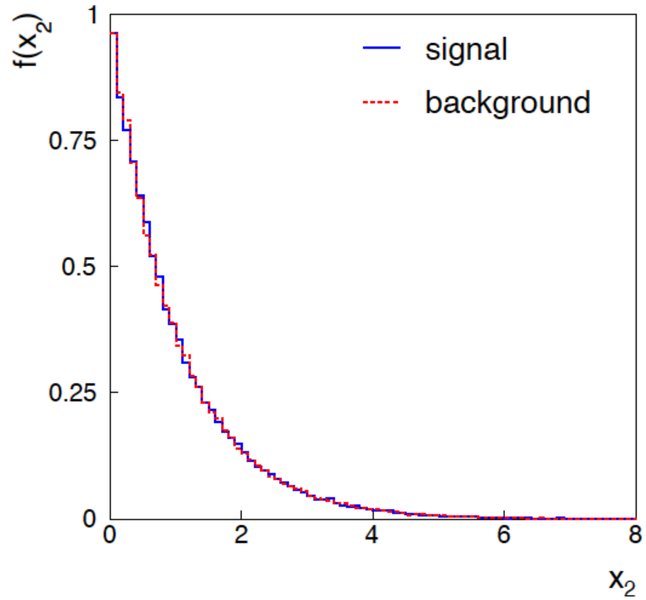
$$f(x_1, x_2|\mathrm{s}) = \frac{1}{\sqrt{2\pi}\sigma(x_2)} e^{-(x_1-\mu_{\mathrm{s}})^2/2\sigma^2(x_2)} \, \frac{1}{\lambda} e^{-x_2/\lambda}$$

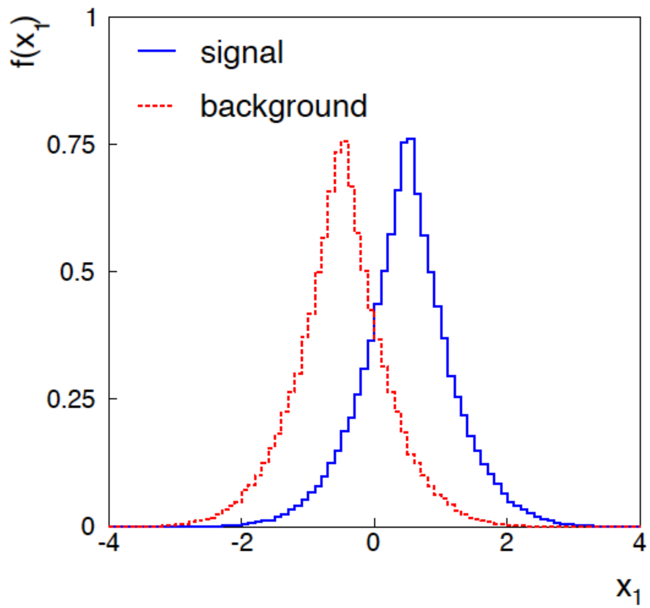
$$f(x_1, x_2|\mathrm{b}) = \frac{1}{\sqrt{2\pi}\sigma(x_2)} e^{-(x_1-\mu_{\mathrm{b}})^2/2\sigma^2(x_2)} \, \frac{1}{\lambda} e^{-x_2/\lambda}$$

$$\sigma(x_2) = \sigma_0 e^{-x_2/\xi}$$

Joint and marginal distributions of x_1, x_2



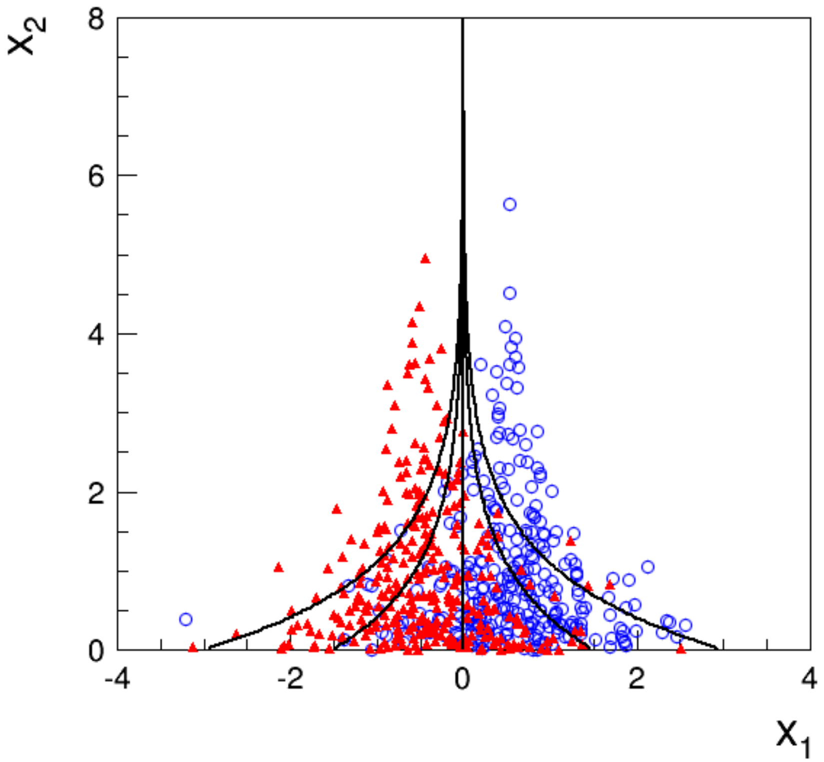




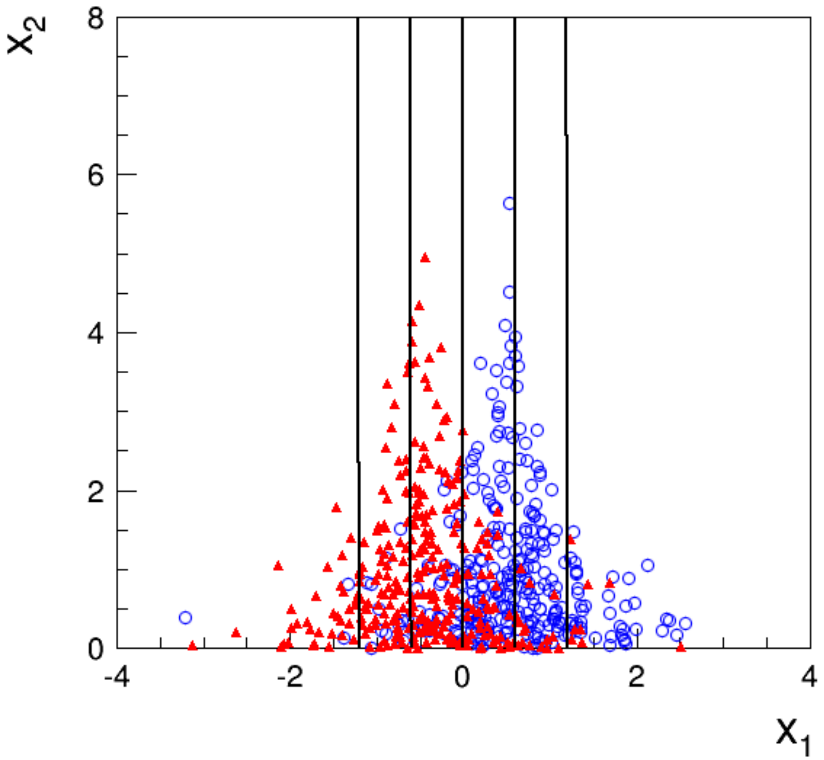
Distribution $f(x_2)$ same for s, b.

So does x_2 help discriminate between the two event types?

Contours of constant decision function

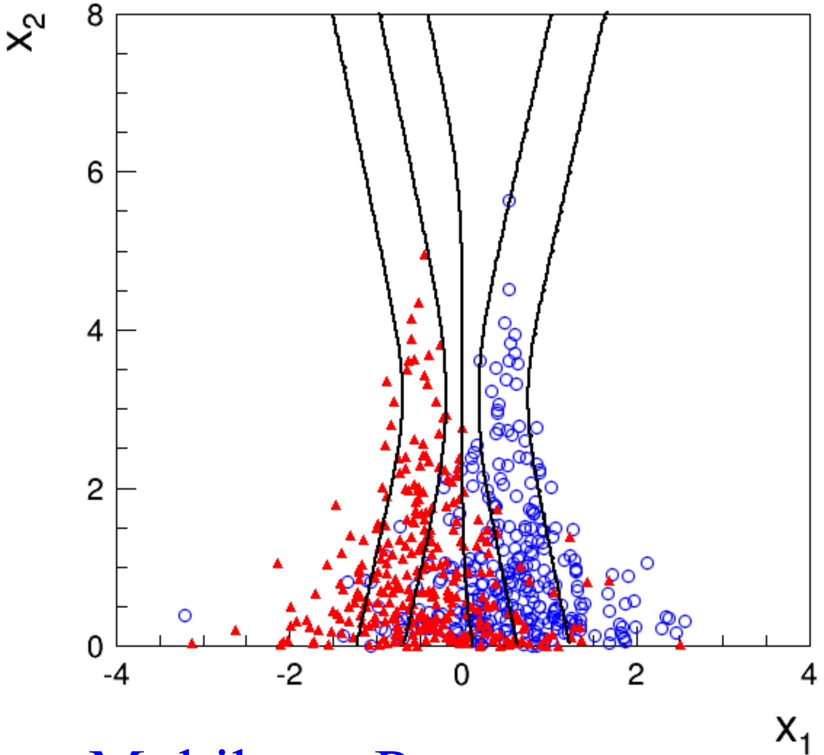


Exact likelihood ratio
(theoretical optimum)

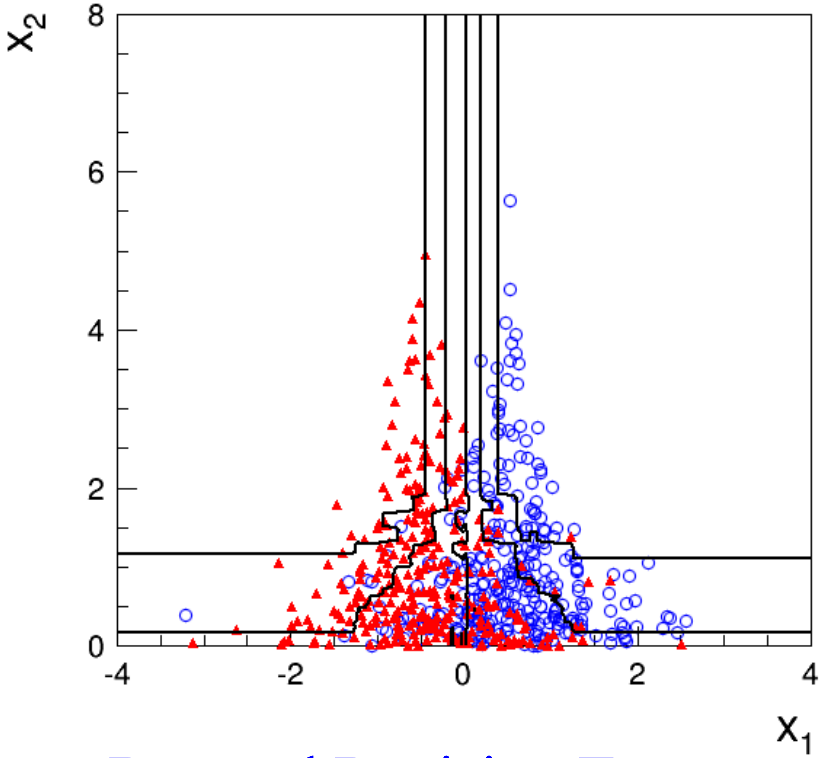


Linear boundary

Contours of constant decision function (2)



Multilayer Perceptron
1 hidden layer with 2 nodes



Boosted Decision Tree
200 iterations (AdaBoost)

Training samples: 10^5 signal and 10^5 background events