Machine Learning

Sergei Gleyzer

PART I

TAE 2017 Lectures

Sep. 4, 2017

Outline

- **What is Machine Learning**
- **in Particle Physics**
- **in Theory**
- **in Practice**

Machine Learning Basics

Machine Learning

What is Machine Learning?

- Study of algorithms that
  improve their performance **P**
  for a given task **T**
  with more experience **E**

Sample tasks: identifying faces, Higgs bosons

Machine Learning

General Approach:

Given **training** data $T_D = \{y, \mathbf{x}\} = (y,x)_1 \ldots (y,x)_N$,

function space {f} and a
constraint on these functions

Teach a machine to learn the **mapping** $y = f(x)$

In Computer Science

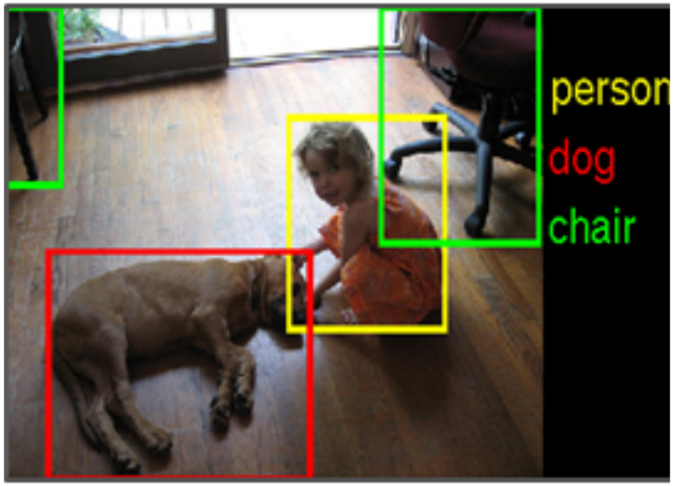
Already the preferred approach to:

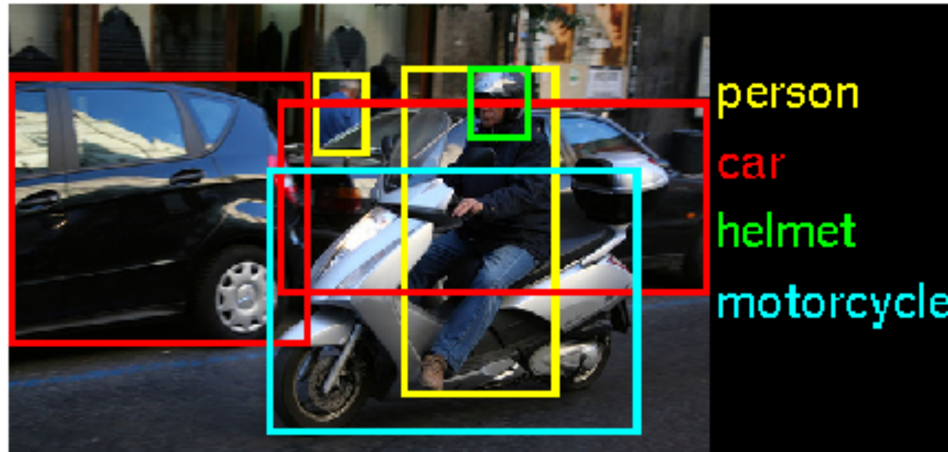
- Speech recognition, natural language processing
- Computer vision, Robot control
- Medical outcomes analysis

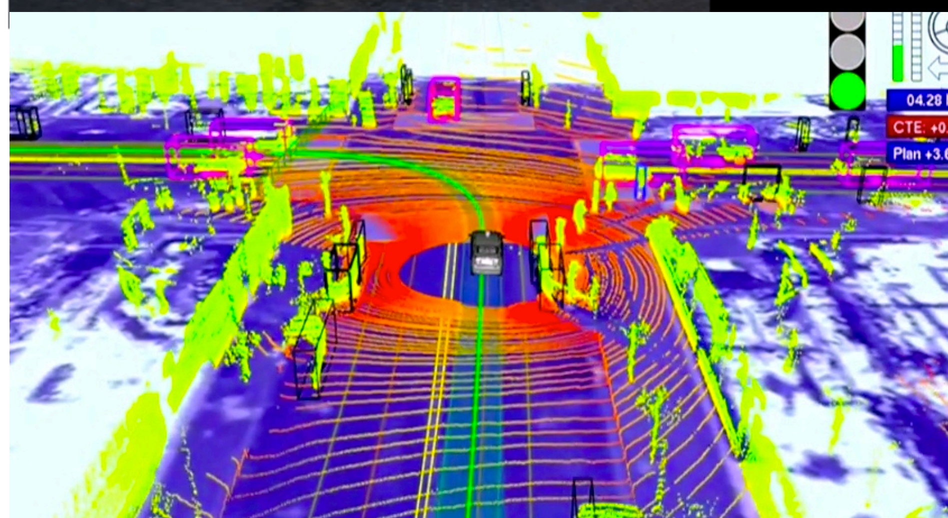
Growing fast

- Improved algorithms
- Increased data capture
- Software too complex to write by hand

Examples







Machine Learning

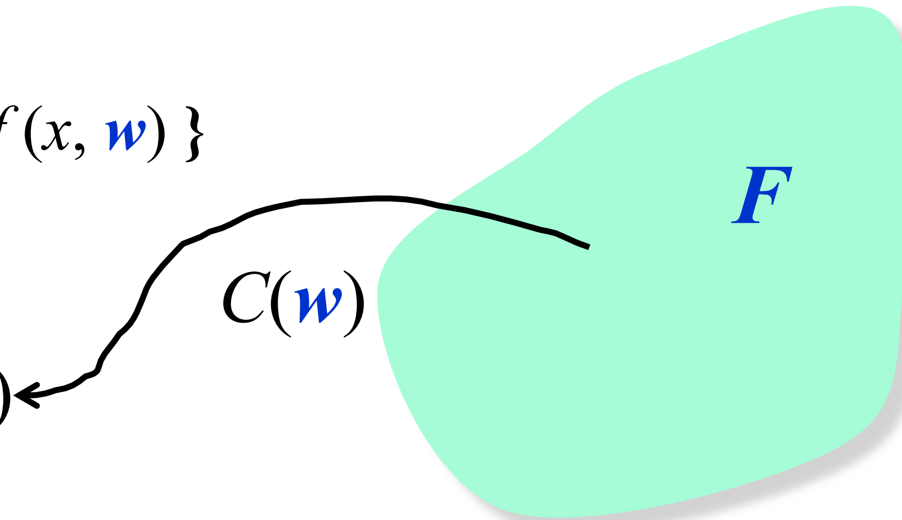
Choose

Function space $\boldsymbol{F} = \{ f(x, \boldsymbol{w}) \}$

Constraint $\boldsymbol{C}$

Loss function* $\boldsymbol{L}$

$f(x, \boldsymbol{w}^*)$



Method

Find $f(x)$ by minimizing the empirical risk $R(w)$

$$R[f_w] = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i, w))$$

subject to the constraint $C(\boldsymbol{w})$

*The loss function measures the cost of choosing badly

Machine Learning

Many methods (e.g., neural networks, boosted decision trees, rule-based systems, random forests,…) use the

quadratic loss

$$L(y, f(x, w)) = [y - f(x, w)]^2$$

and choose $f(x, w^*)$ by minimizing the

constrained mean square empirical risk

$$R[f_w] = \frac{1}{N}\sum_{i=1}^{N}[y_i - f(x_i, w)]^2 + C(w)$$

History

1950s: First methods invented

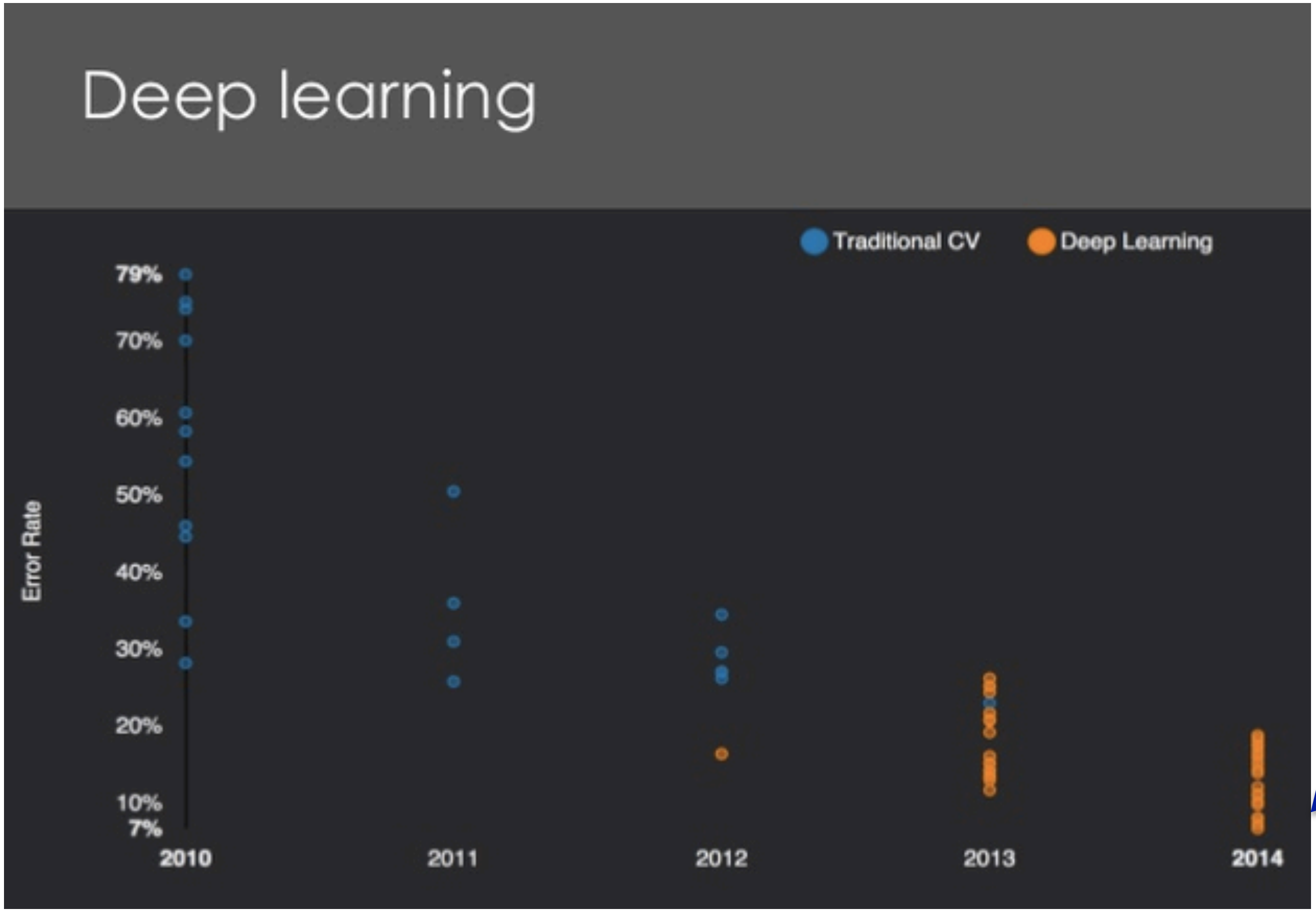
1960-80s: Slow growth, focus on knowledge

1990s: Growth of computing power, new learning methods, data-centric

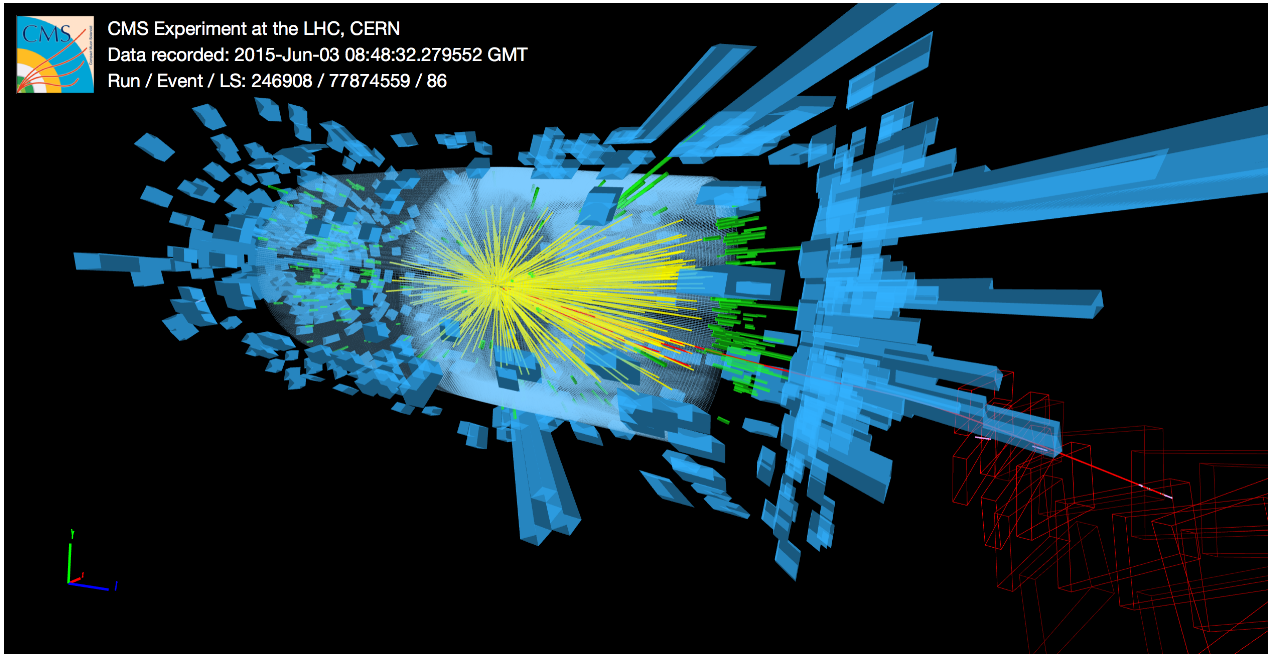
2000-10s: Wider use in research and industry

2010s: Learning improvement, dedicated hardware, deeper learning

Diving Deeper



Huge Progress

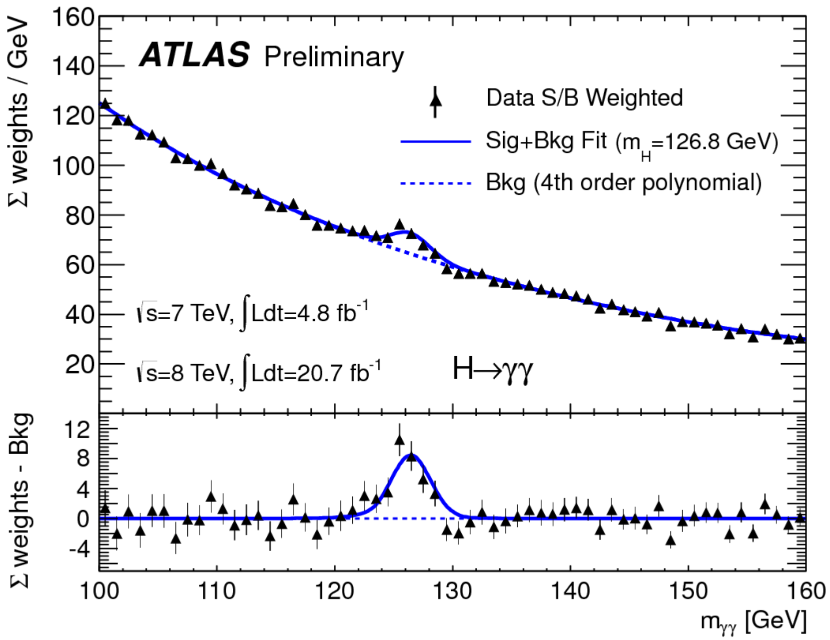


In Particle Physics

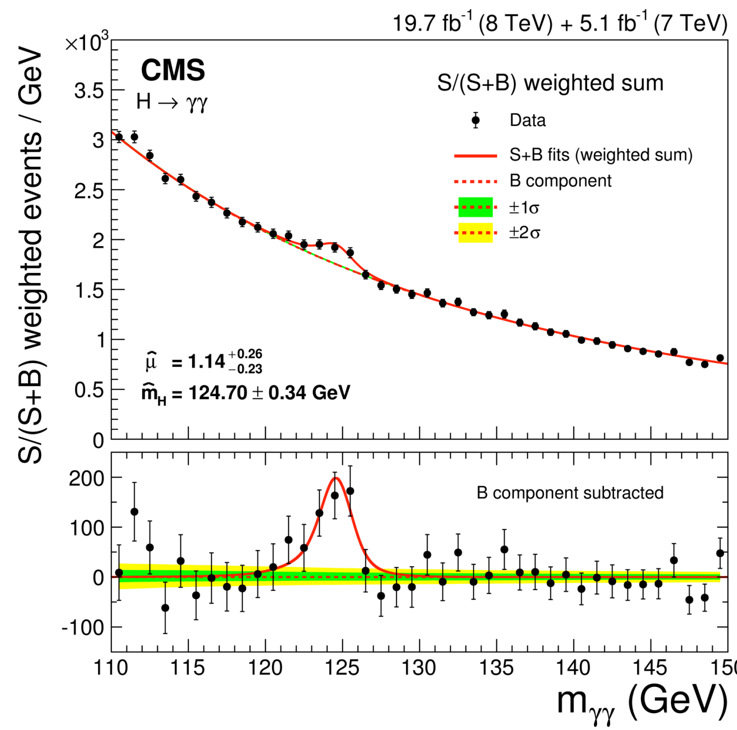
1

Higgs Boson Discovery

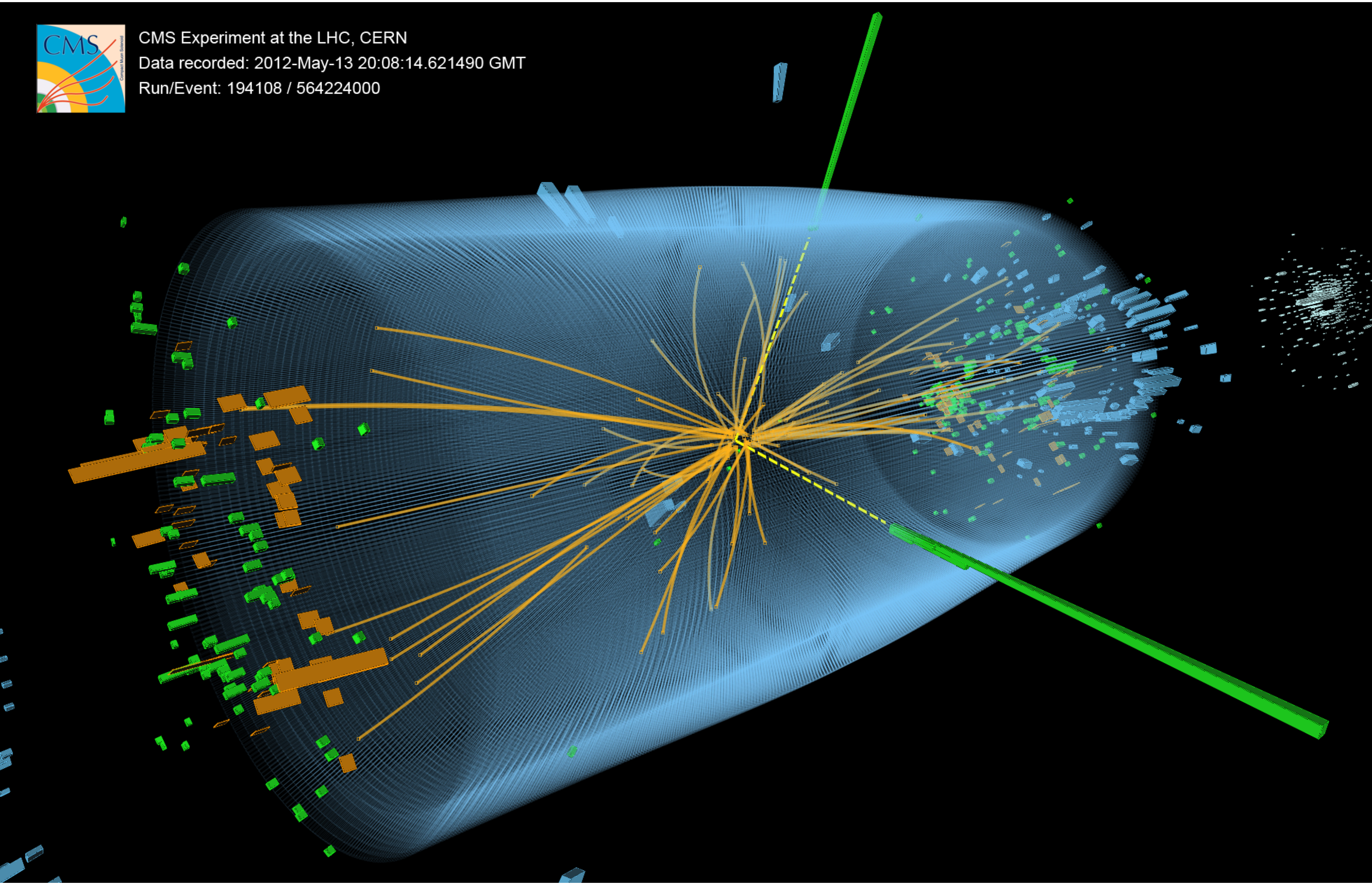
Higgs to di-photons



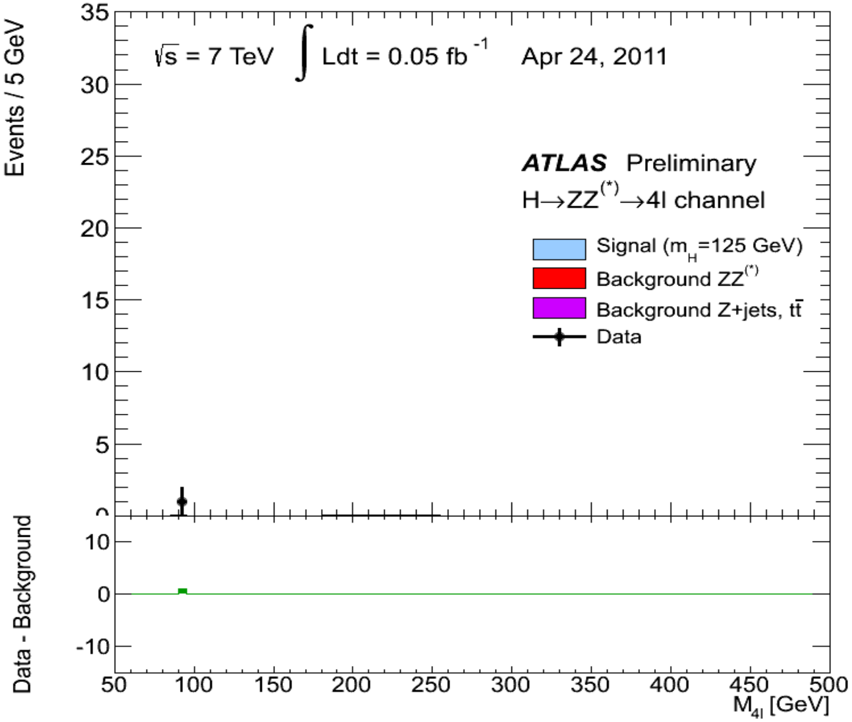
ATLAS

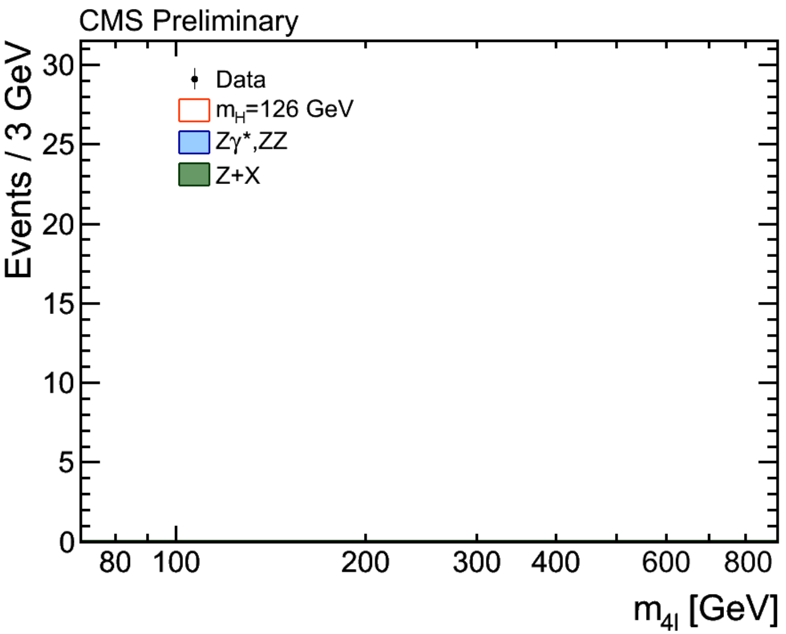


CMS


CMS
CMS Experiment at the LHC, CERN
Data recorded: 2012-May-13 20:08:14.621490 GMT
Run/Event: 194108 / 564224000

Higgs → 4 leptons



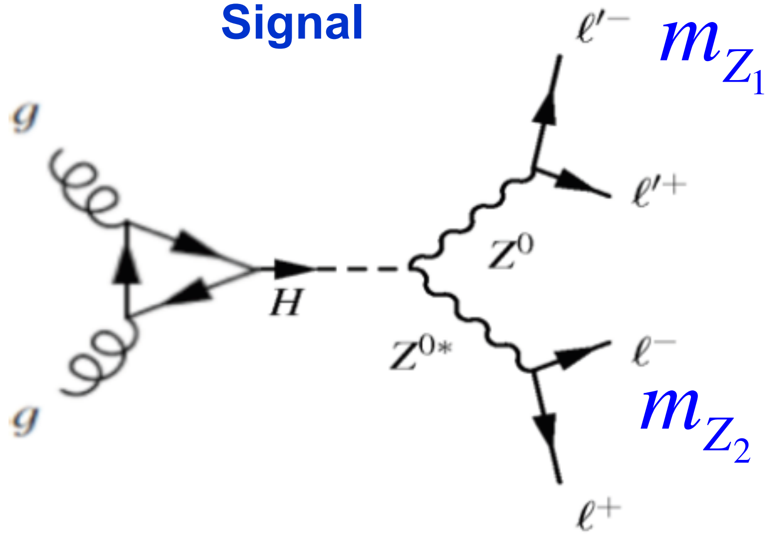


ATLAS

CMS

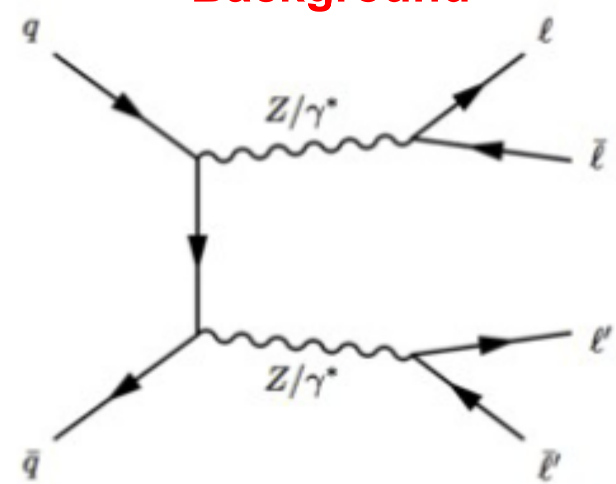
Higgs → 4 leptons

Signal



$$pp \rightarrow H \rightarrow ZZ \rightarrow \ell^+\ell^-\ell'^+\ell'^-$$

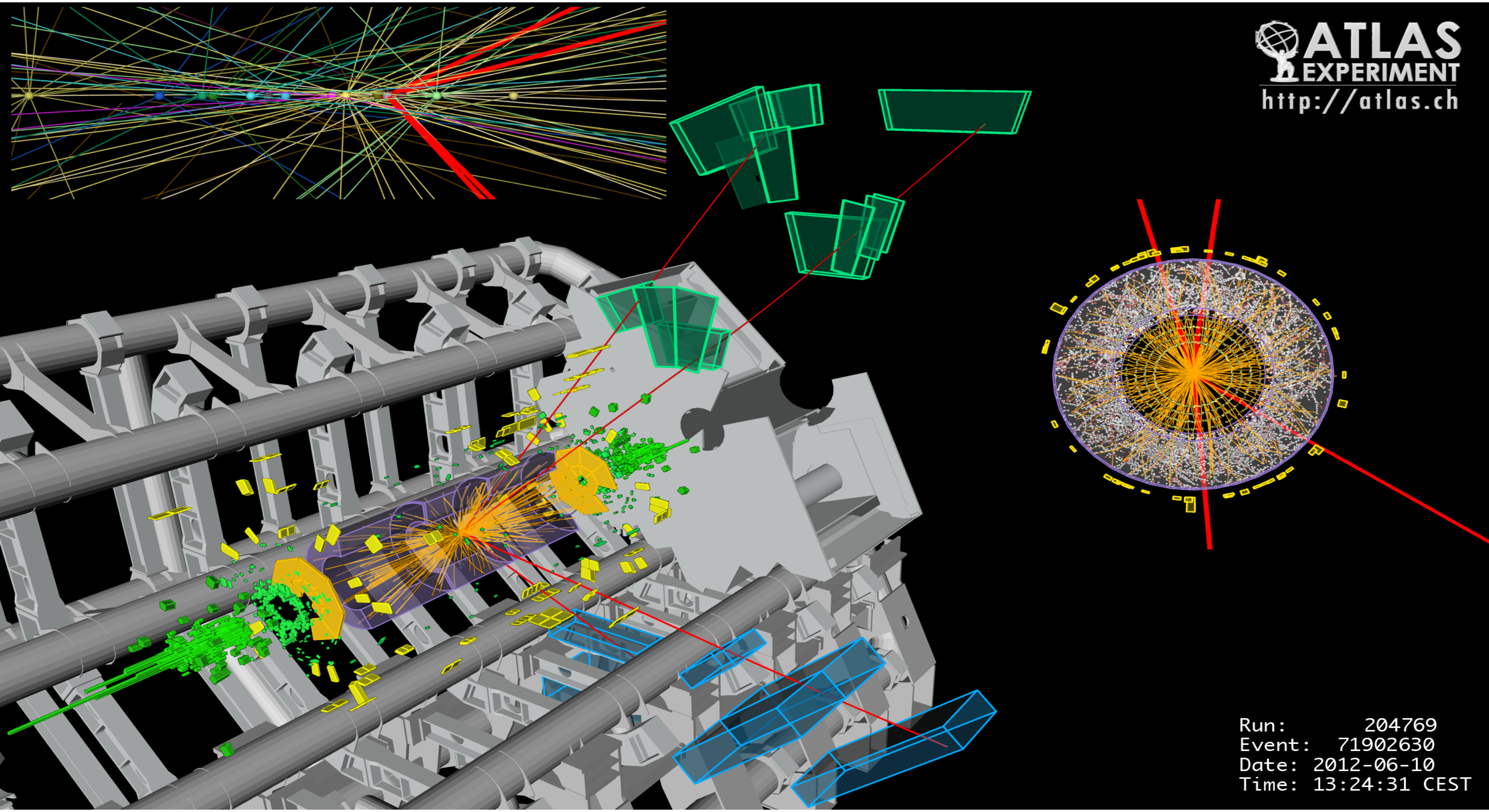
Background



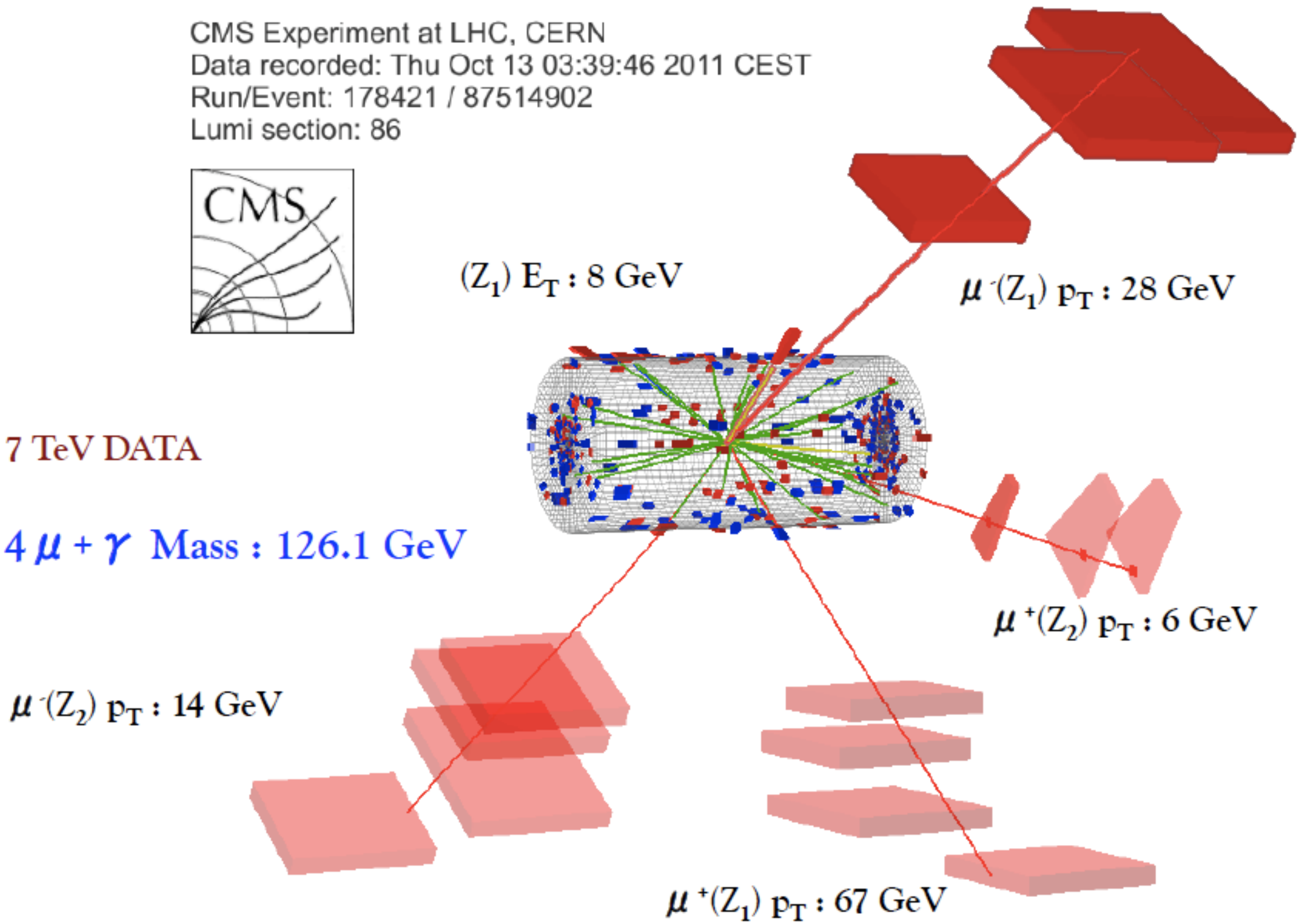
$$pp \rightarrow ZZ \rightarrow \ell^+\ell^-\ell'^+\ell'^-$$

$$\mathbf{x = (m_{Z1}, m_{Z2})}$$

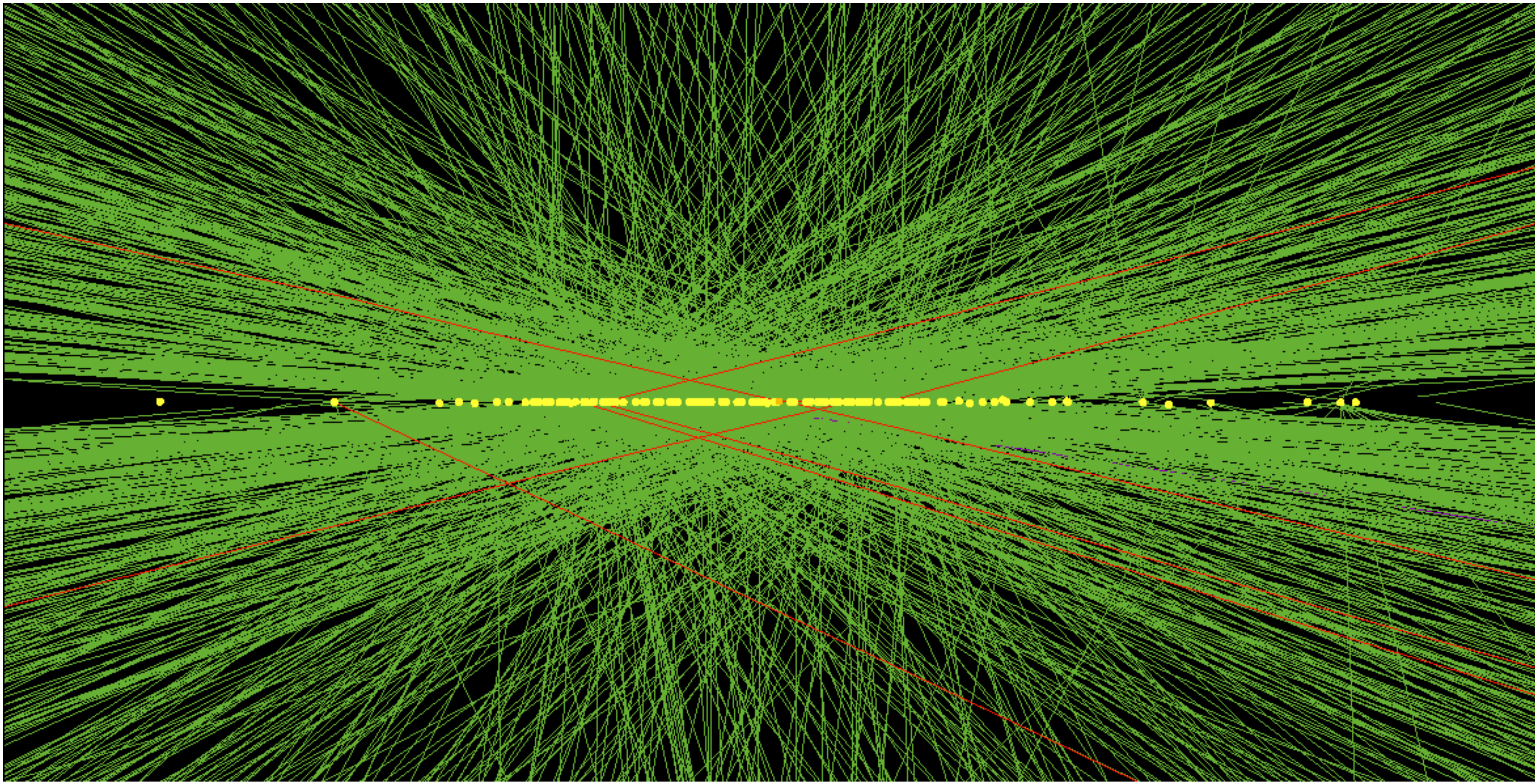
4-lepton event ATLAS



4-lepton event CMS

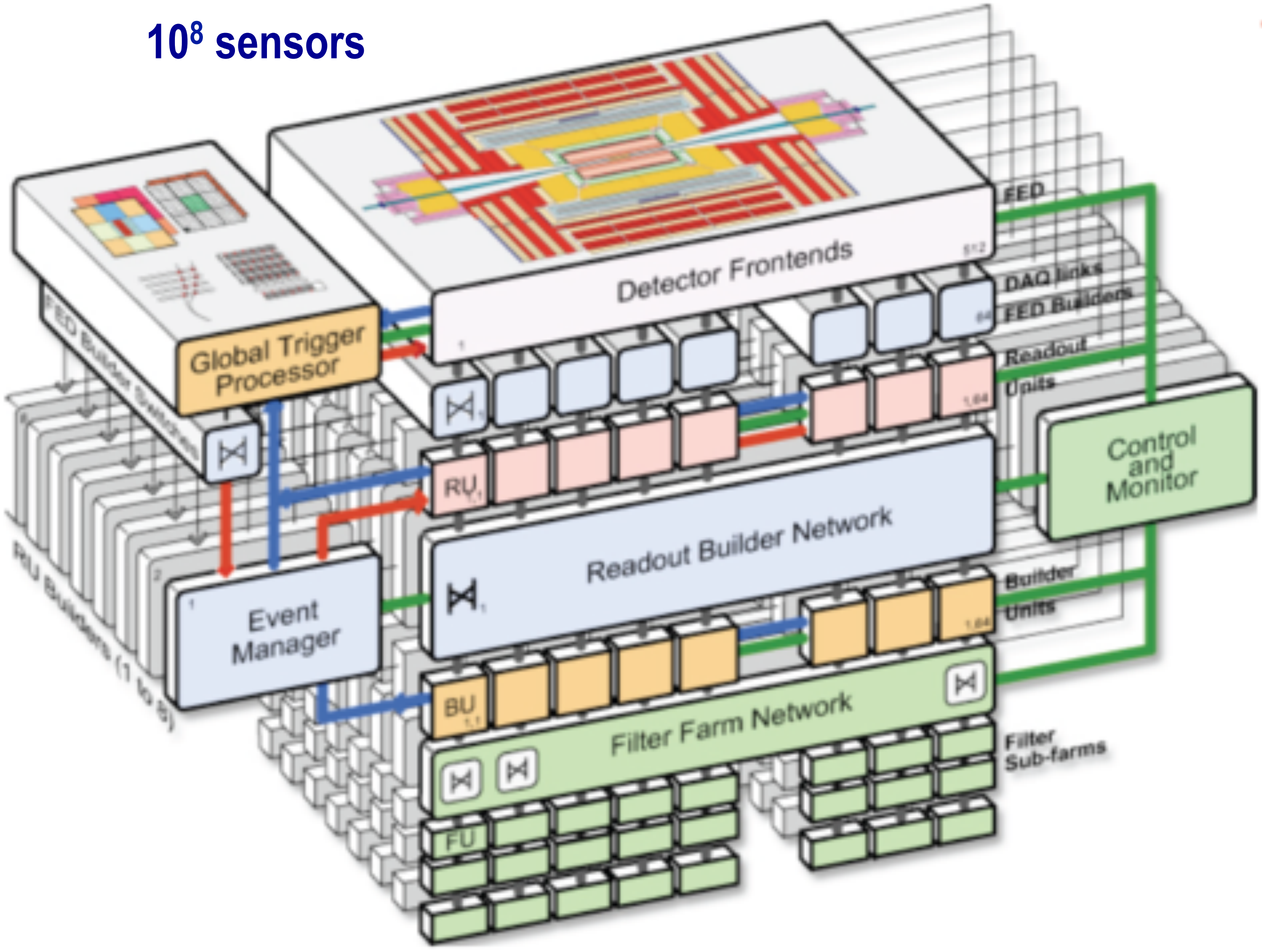


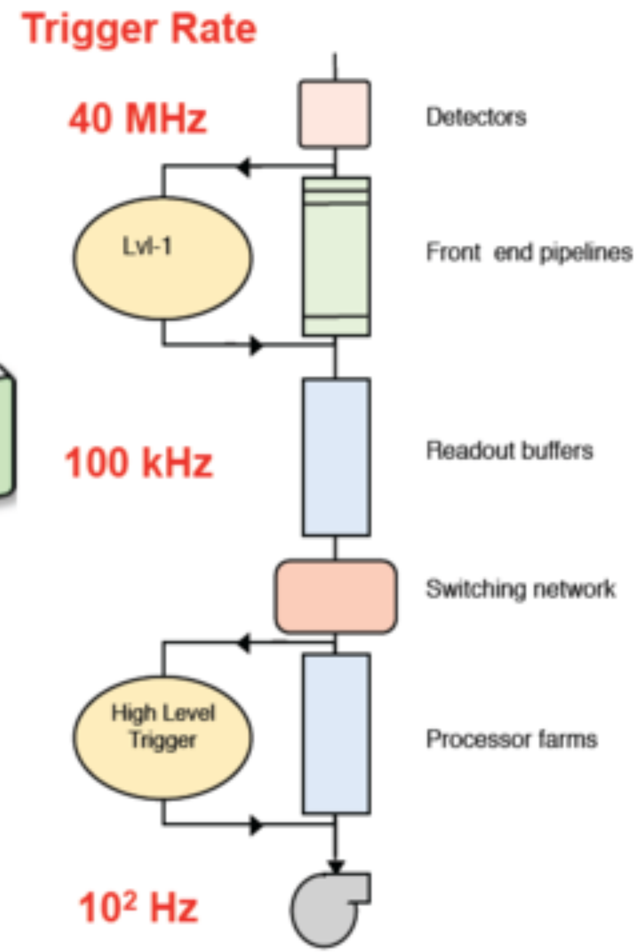
Event Complexity



Event Filtering

10^8 sensors

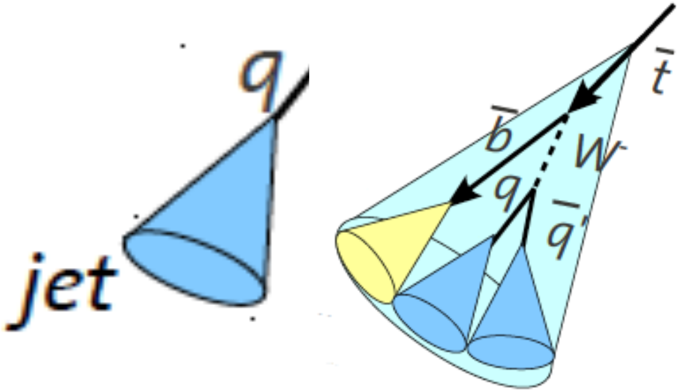




Applications

I. Classification

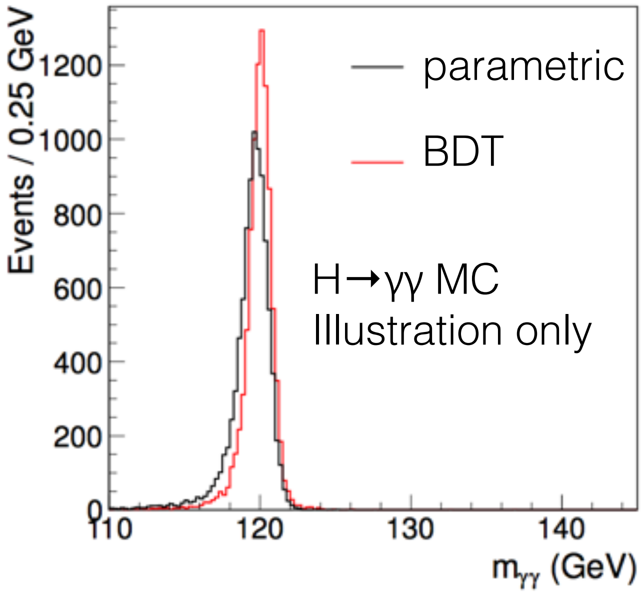
- Particle Identification
- Pattern Recognition (tracks)
- Searches for New Physics
- Data Quality Monitoring



Applications

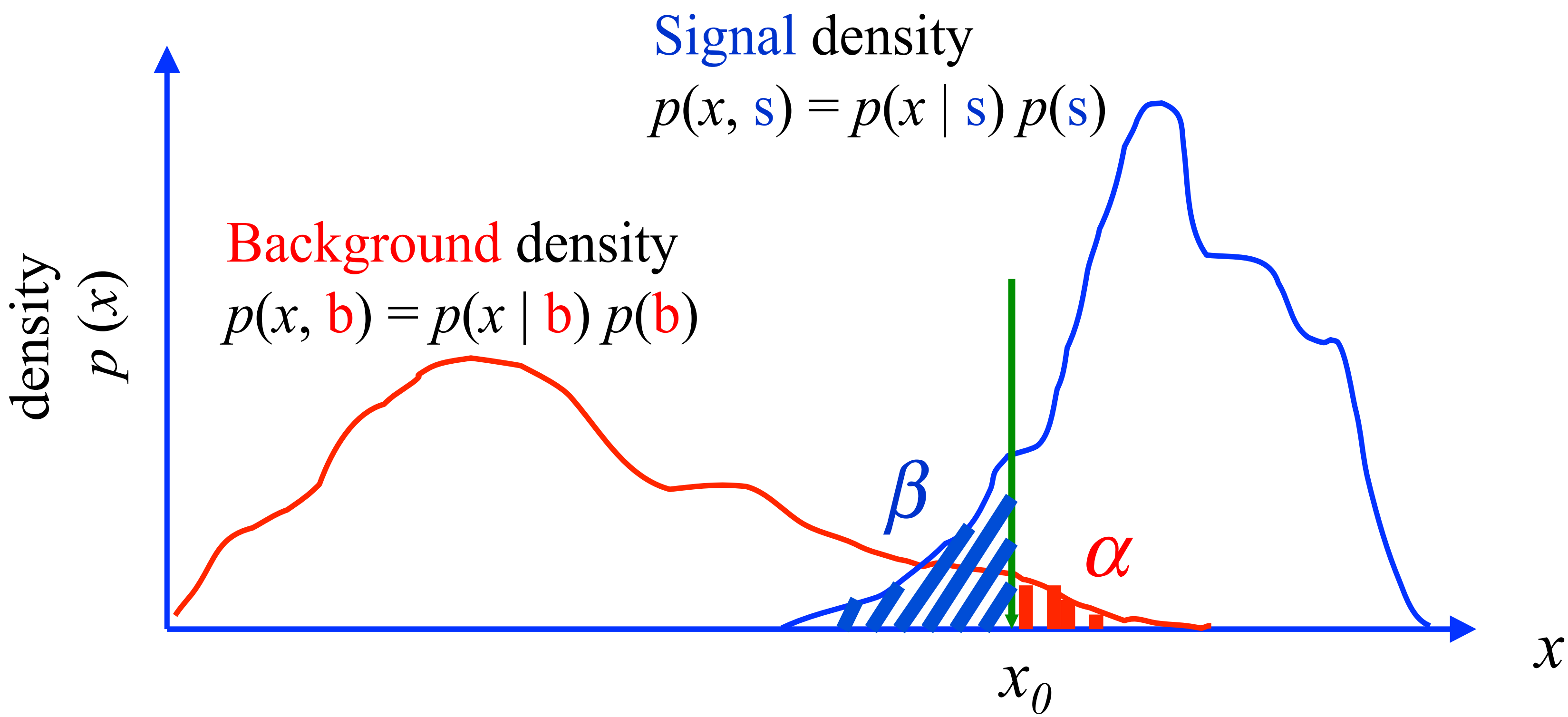
II. Function estimation

- Particle energy better estimated with ML methods
- ML Regression



Classification Theory

Classification Theory



Optimality criterion: minimize the error rate, $\alpha + \beta$

Classification Theory

The total loss L arising from classification errors is given by

$$L = L_b \int H(f)\, p(x,b)\, dx + L_s \int [1 - H(f)]\, p(x,s)\, dx$$

Cost of background misclassification

Cost of signal misclassification

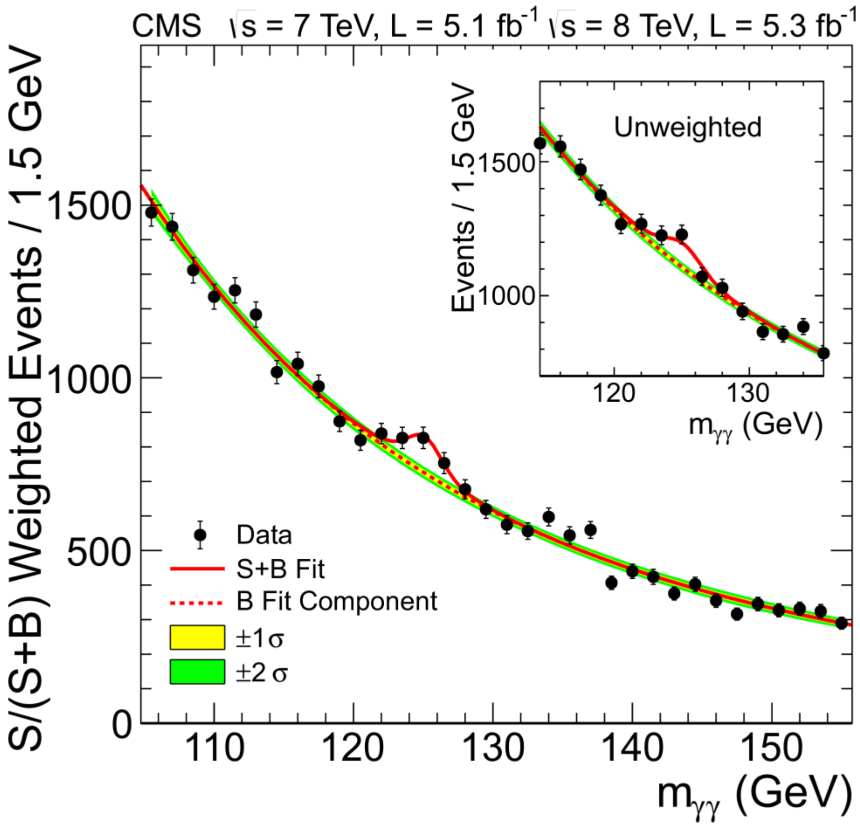
where $f(x) = 0$ defines a decision boundary such that $f(x) > 0$ defines the acceptance region

$H(f)$ is the Heaviside step function:

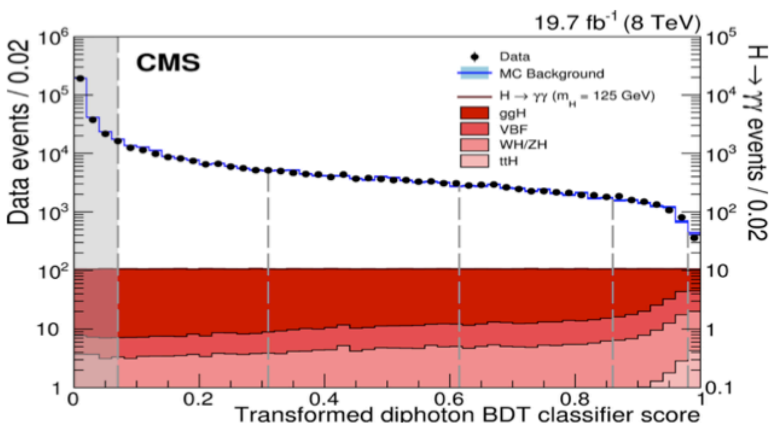
$H(f) = 1$ if f > 0, 0 otherwise

Classification in Practice

in Higgs Discovery

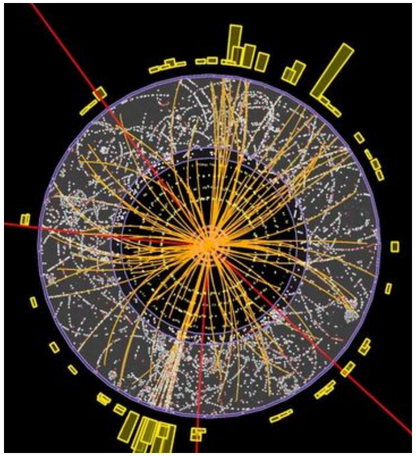


- Identification of particles
- Identification of interactions
- Energy regression
- Event selection

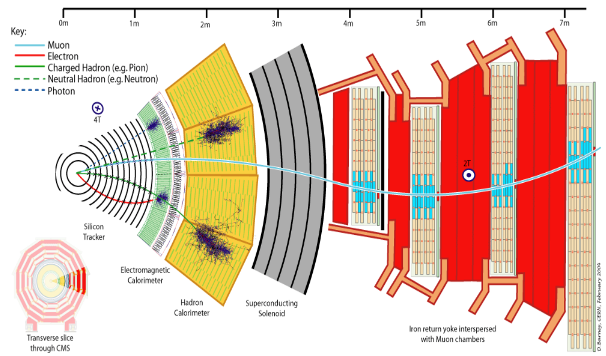


Improvement in analysis from all four areas

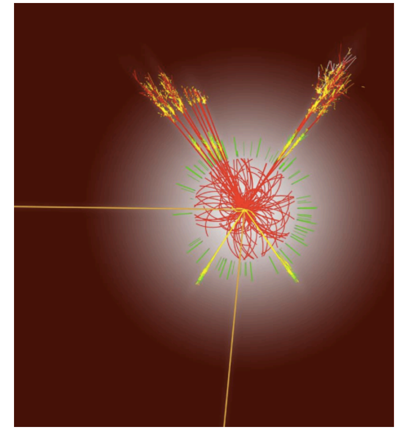
Interesting areas



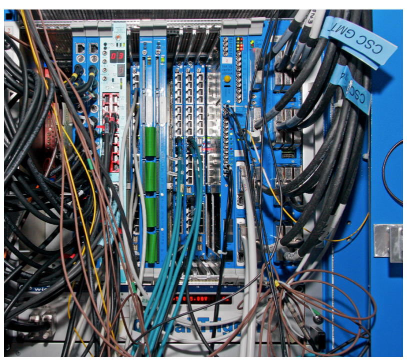
Tracking



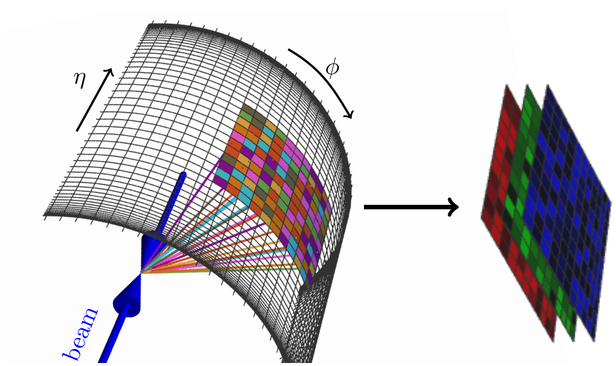
Fast Simulation



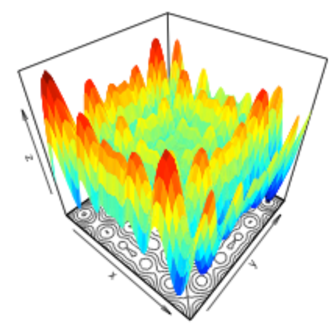
Object Identification



Event Filtering



Imaging Techniques



Simulation

CONSTRUCTING CLASSIFIERS

Classification

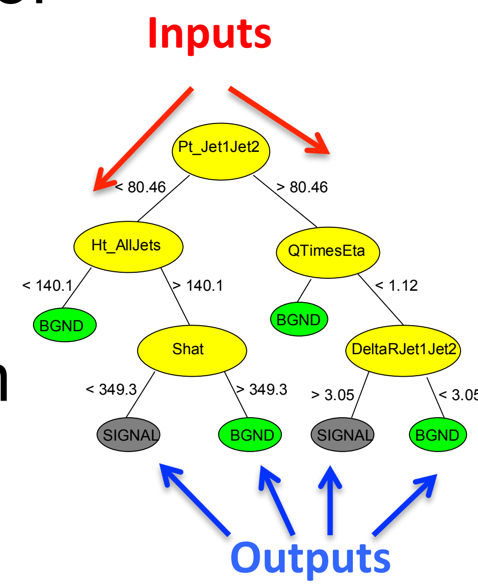
Distinguish f(x), **g(x)** using Training set of observations

{**inputs** , **outputs**}

Pass observations to a learning algorithm
neural network, decision tree

that produces **outputs** in response to **inputs**

Use another set of observations to evaluate



Classification

Primary Goal:

Achieve **lowest probability** of error on unseen cases $\{<x^{(i)}, y^{(i)}>\}$

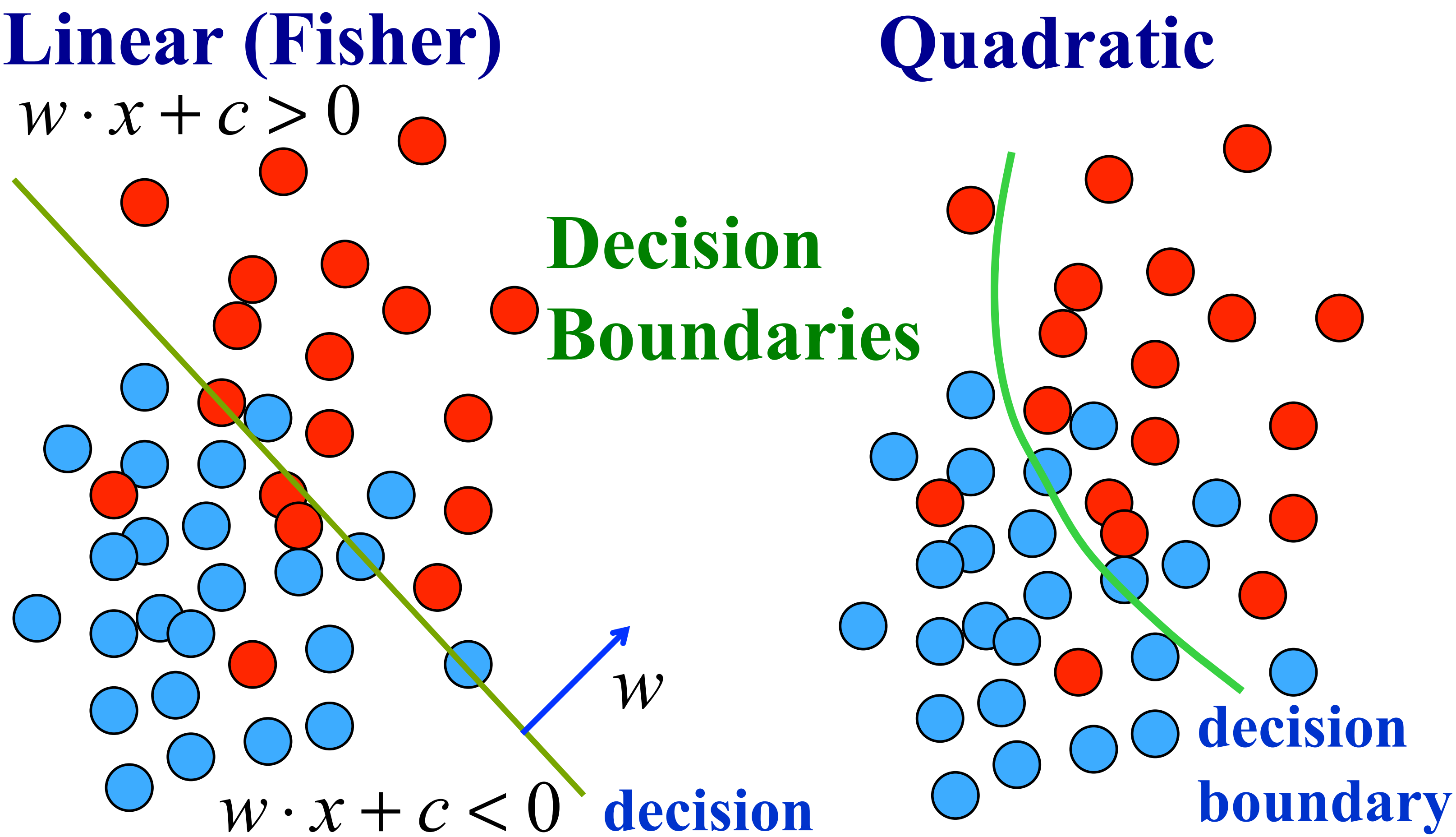
Approach:

Inductively learn from labeled examples (where classes are known)

ML Algorithms

- **Fisher, Quadratic**
- **Naïve Bayes (Likelihood)**
- **Kernel Density Estimation**
- **Random Grid Search**
- **Rule ensembles**
- **Boosted decision trees**
- **Random forests**
- **Deep learning neural networks**
- **Support vector machines**
- **Genetic algorithms**

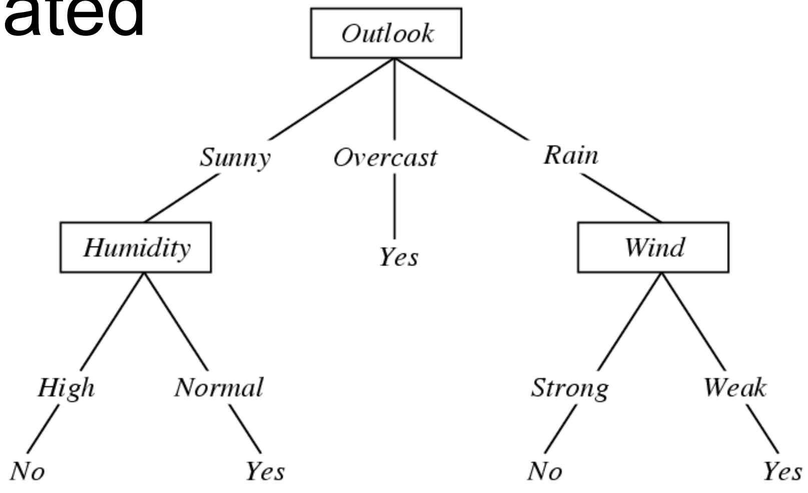
Linear and Quadratic



Binary Decision Trees

Decision Trees

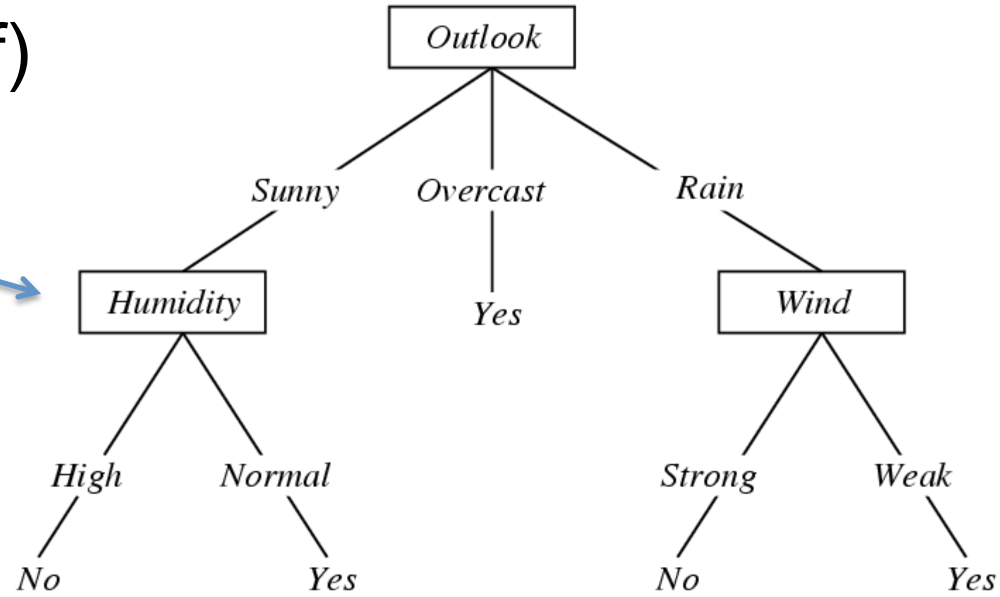
- **Decision trees** are **multidimensional histograms**
  - Recursively constructed bins
  - Each associated to the value (or **class**) of f(x) to be approximated
  - **Golf-Playing** Decision Tree: f(outlook, humidity, wind, T)



Decision Trees

- Each **internal** node: test one attribute X_i
- Each **branch**: selects one value for X_i
- Each **leaf** node: predict Y
  - Or P(Y|X in leaf)

Decision Node



Decision Tree Learning

- Unknown **target function** f: **X→Y**
  - **Y** is discrete valued (class)
- Set of possible instances **X**
  - each **instance** is a feature vector

e.g. <Humidity = High, Wind = weak, Outlook = rain, Temp = hot>

Decision Tree Learning

Input:

– Training examples {<x^i,y^i>}

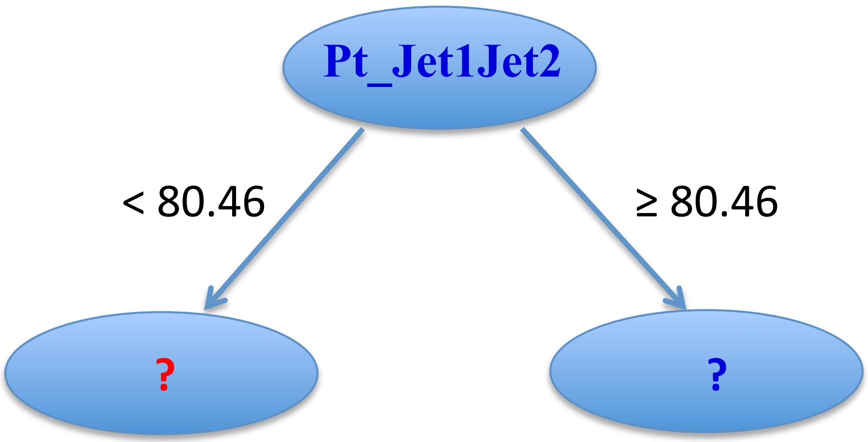
Output

– Hypothesis $h \in H$ that
  best approximates target function f
– Tree sorts x to leaf, which assigns y

Decision Trees

Building a tree:

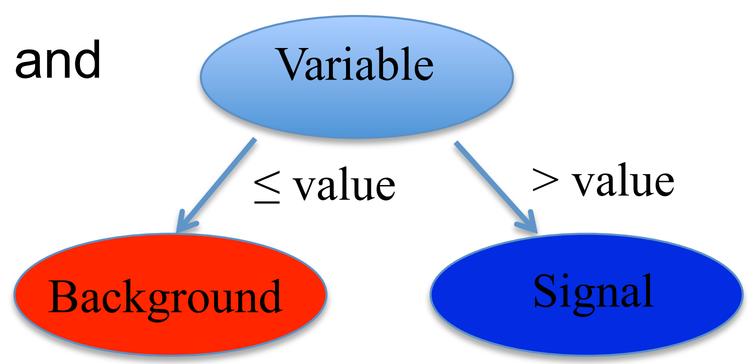
- Scan along each variable and propose a **DECISION**
  - A cut on value that maximizes class separation (binary branching)



Decision Trees

- Choose **decision** that leads to greatest separation among classes **signal**/ **background**
  - Based on the information gained from split
    - Build regions of increasing purity
    - Stop when no further improvement from additional branching
    - Reach terminal node (leaf) and assign purity-based class

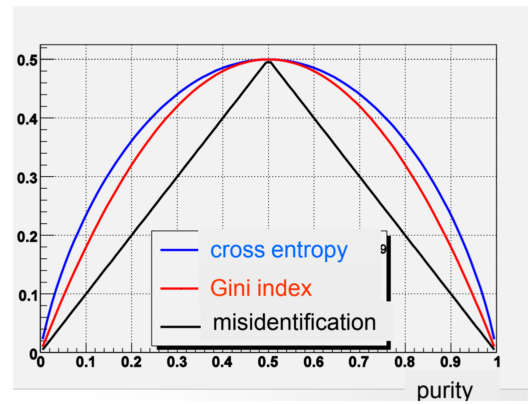
$$\frac{N_{signal}}{N_{signal} + N_{background}}$$

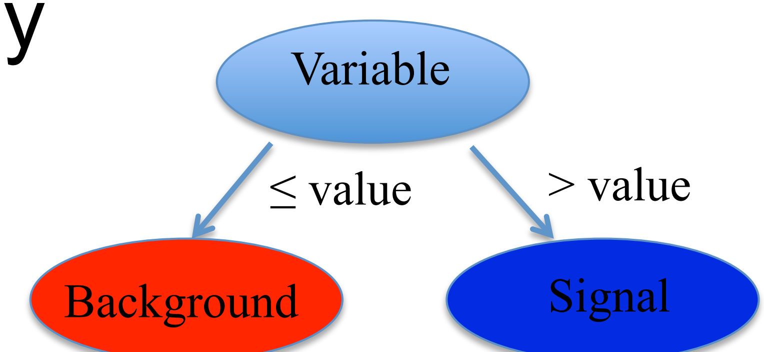


Separation Gain

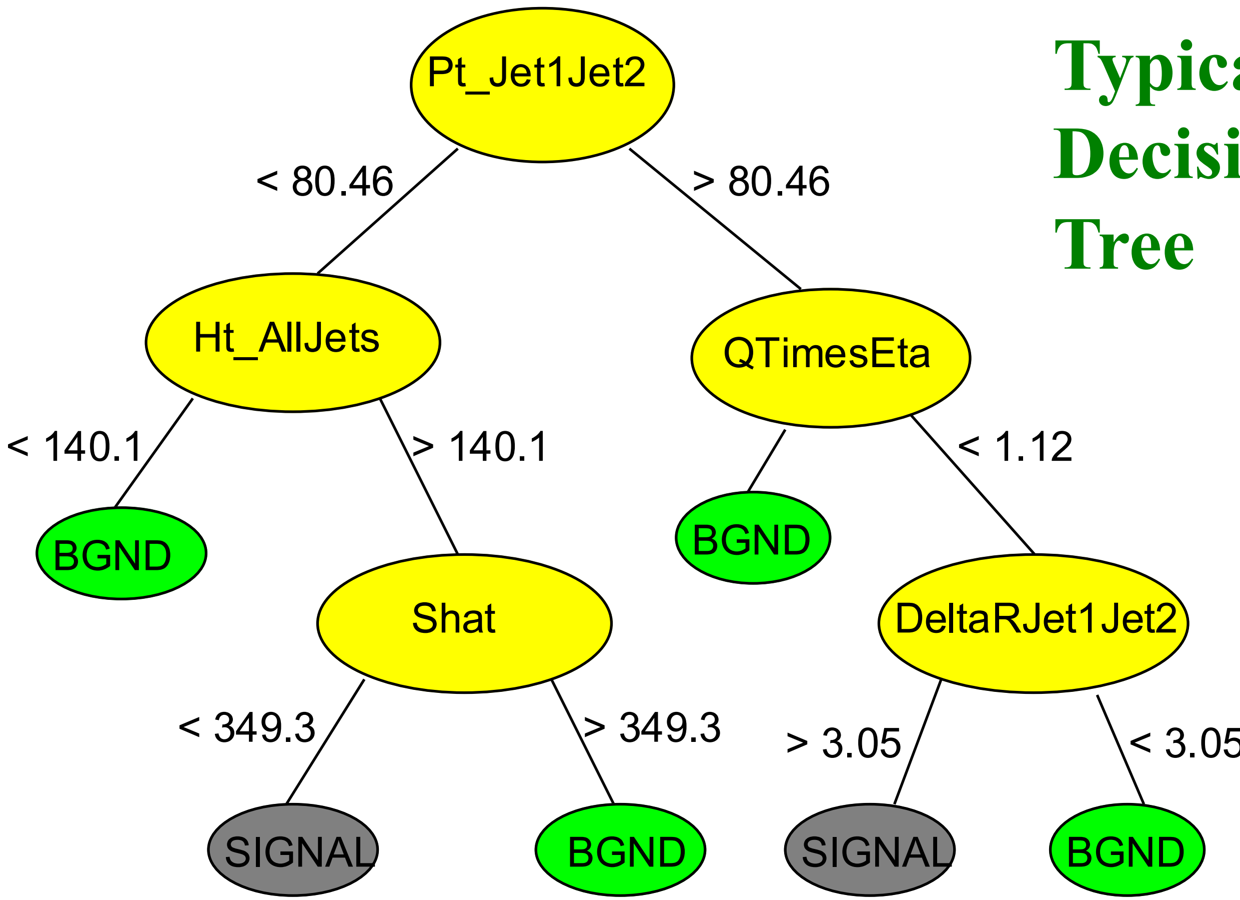
Measures of Separation Gain

- Cross-Entropy
  - -(plnp + (1-p)ln(1-p)
- Gini Index
  - p (1 – p)
- Want to lower entropy due to split





Representation



Typical Decision Tree

Pruning

Decision trees can become large and complex and risk over-fitting the data

Pruning: remove parts of the tree that are less powerful or possibly noisy

- start from the leaves and work back up

Pruned trees smaller in size, easier to interpret

Summary

- **Machine Learning is a very powerful field with an expanding number of applications**
  – Basic Methods: Linear, Quadratic, Decision Trees, Decision Rules
  – More advanced methods next time
  – Many methods available, good to experiment