Parallelization of ROOT Machine Learning Methods

Pourya Vakilipourtakalou

Supervisors : Prof. Lorenzo Moneta

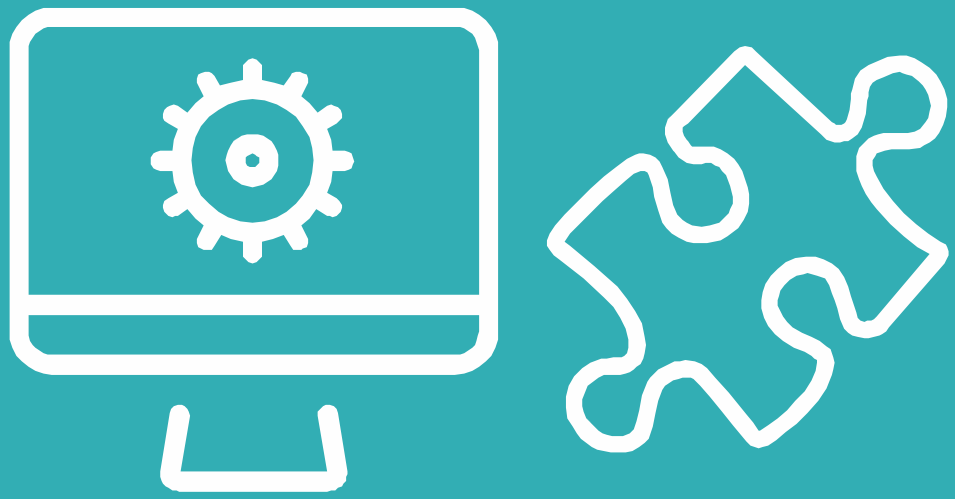
Prof. Sergei Gleyzer

Overview

- Machine Learning
- ROOT
- TMVA
- Cross Validation
- Parallelization
- Outlook

Machine Learning

Teaching the computers to do something exactly like the way people learn.



How do people learn?

Horse:



Horse!

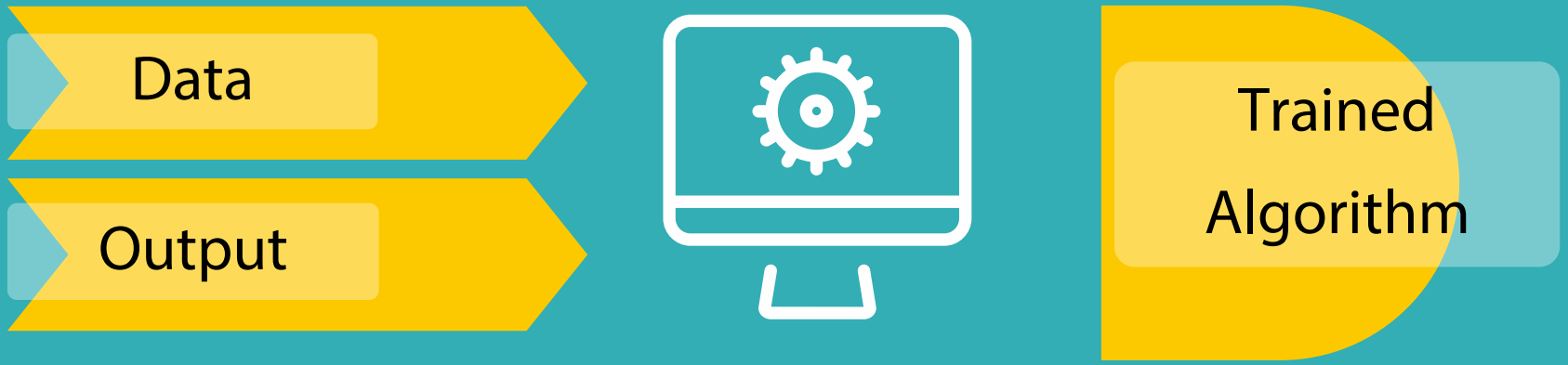
And this is Machine Learning!

Cake:

Cake!

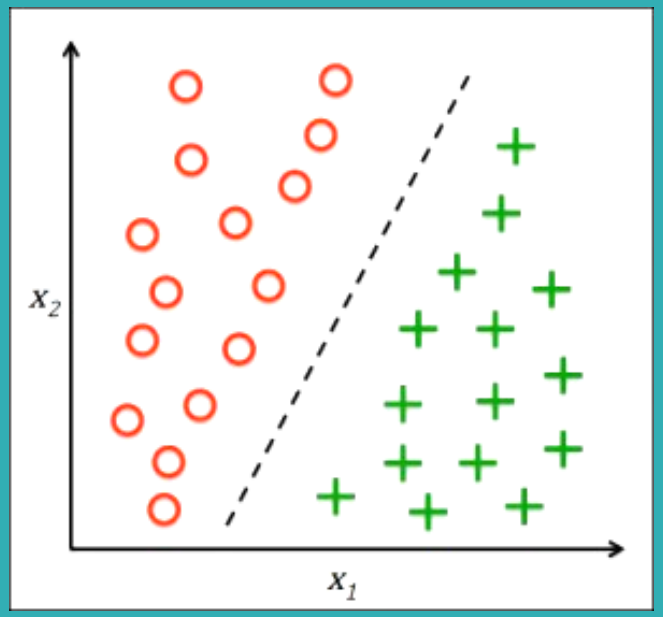
Traditional Programming

Machine Learning



We train the algorithm on known data sets and we want to find the answer for the unknown cases so we ask the computer to do this.

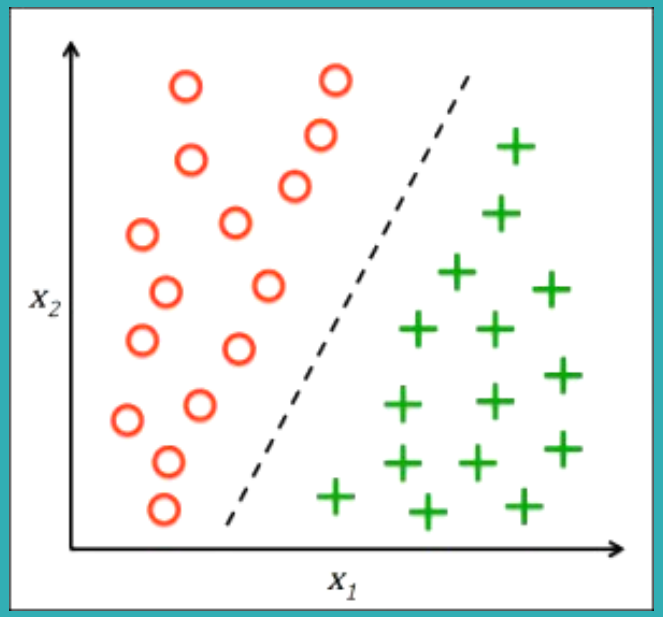
Machine Learning



- An example of classification
- X1 ⟶ age of the patient
- X2 ⟶ size of the tumor
- Y ⟶ output : Malignant or Benign ⟶ 0 or 1
- Proposing a function like H(X1,X2) like

  aX1 + bX2 ⟶ it can be anything
- Try to find optimal a and b

Machine Learning



More Physical Example

- X ⟶ vector of Kinematic Variables
- Y ⟶ output : Higgs (Signal) or Background ⟶ 0 or 1
- Proposing a function like F(X)

modular scientific software framework

mainly written in **C++**

ROOT

functionalities for big data processing and statistical analysis

integrated with other languages **Python** and **R**.

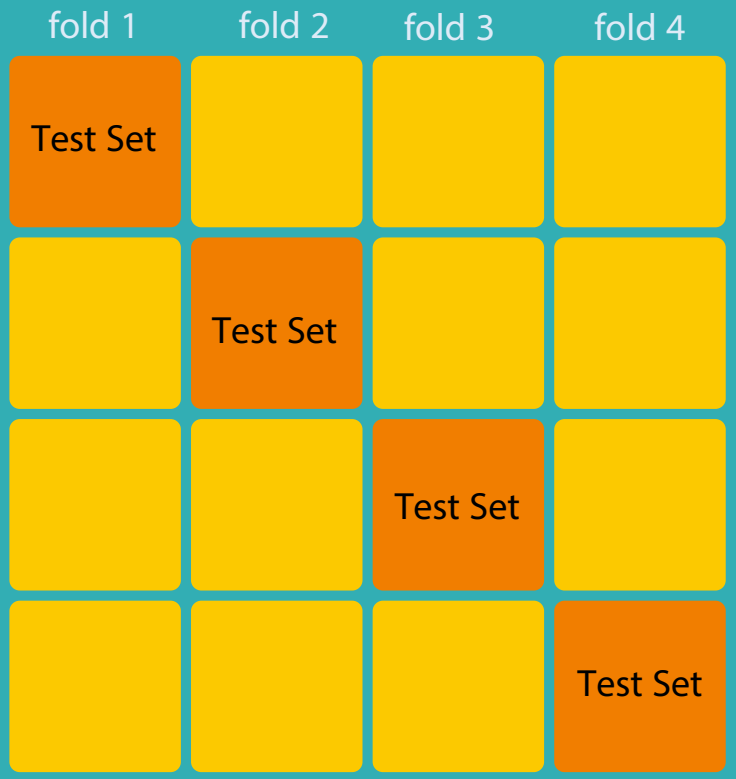
TMVA

ROOT, Machine Learning → TMVA

- Toolkit for Multivariate Data Analysis
- Bunch of methods that provides a ROOT-integrated machine learning environment
- It includes Rectangular cut optimization, Boosted/Bagged decision trees, Artificial neural networks, …

Cross Validation

- Complete dataset
  - Training dataset
  - Test dataset



K-fold cross validation (k=4)

1st iteration → Calc. error

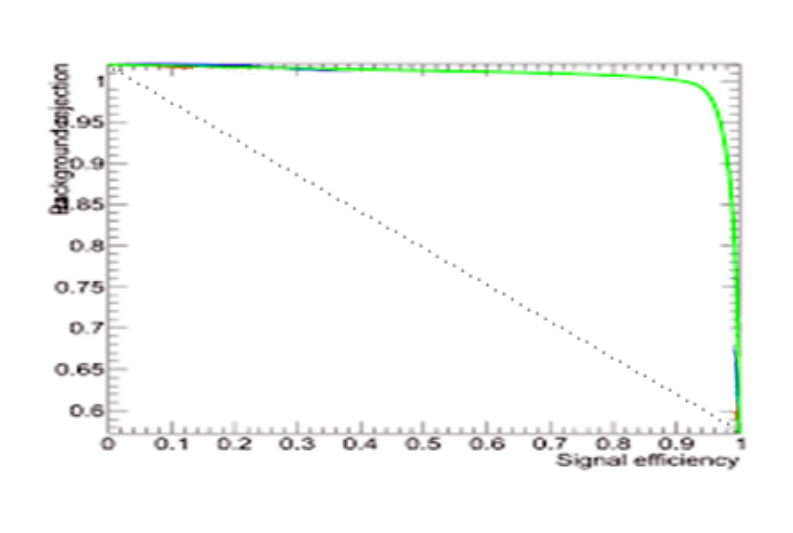
2nd iteration → Calc. error

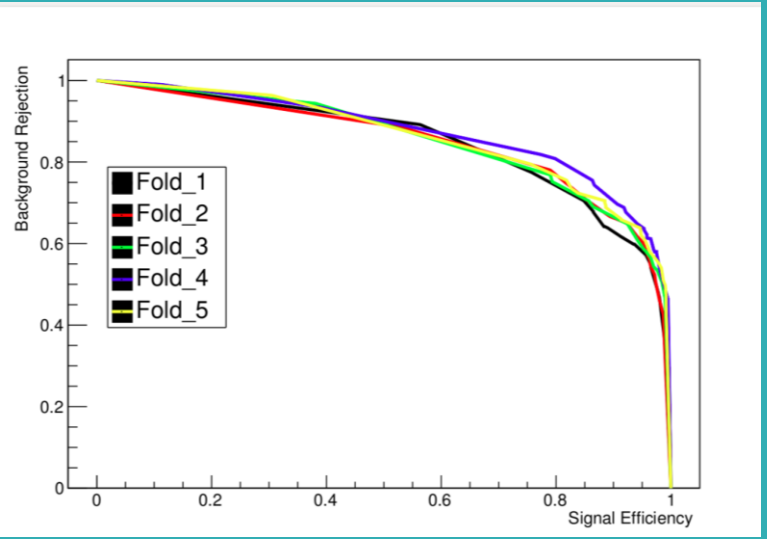
3rd iteration → Calc. error

4th iteration → Calc. error

Calculate avg. error

Cross Validation: PlotROC()





PlotROC() ⟶ Small Higgs Data Set

ROC ⟶Receiver Operating Characteristic: in Statistic graphical plot that illustrates the performance of a classifier

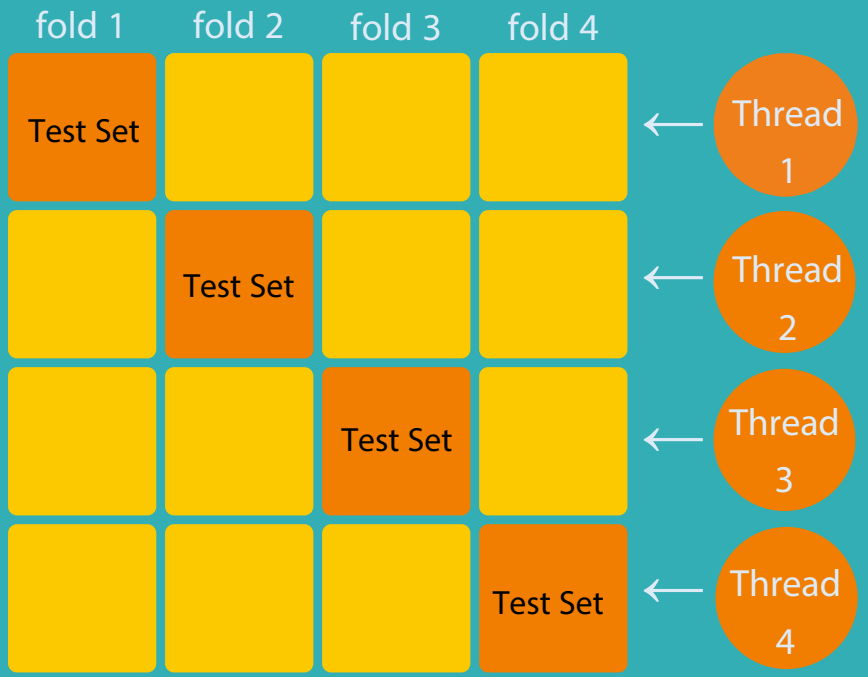
Parallelization

ROOT Classes for Parallelization

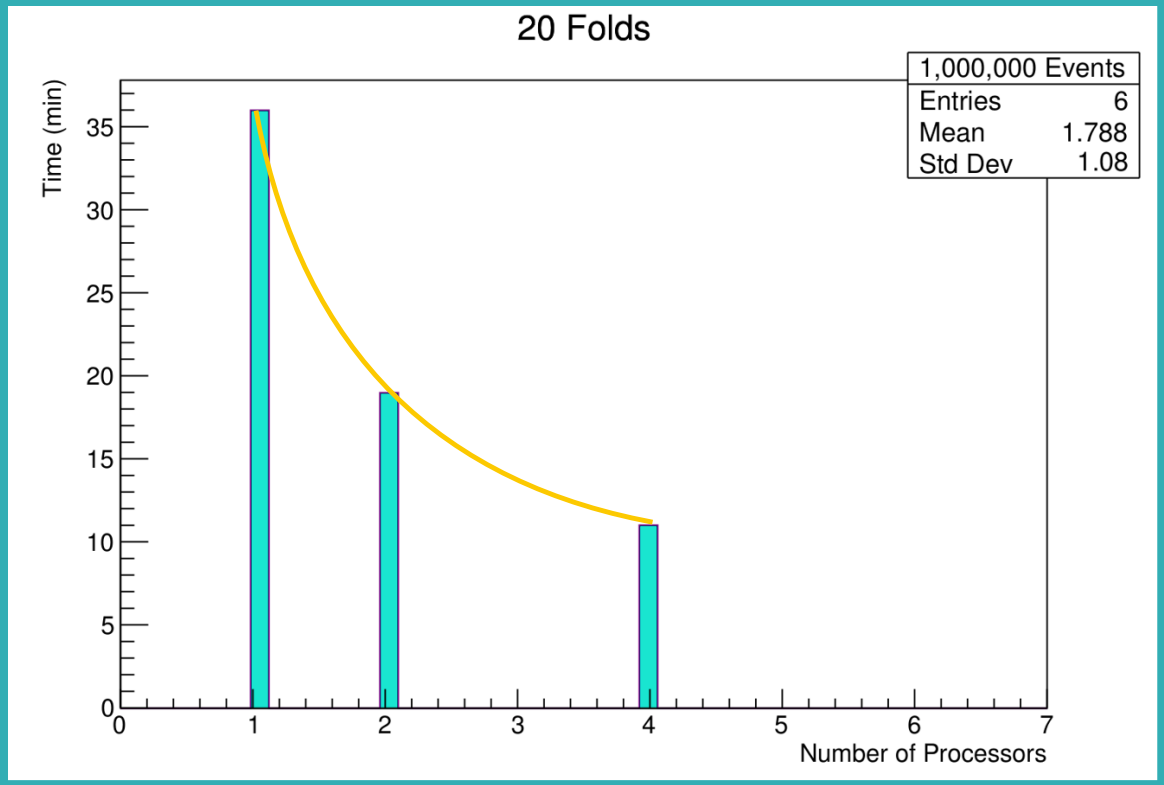
- ThreadPool ⟶Multithreading
- TProcPool ⟶Multiprocessing

- Multithreading ⟶More difficult to Implement : needs locking ⟶ no Global Variable

- Multiprocessing is easier but in some cases slower



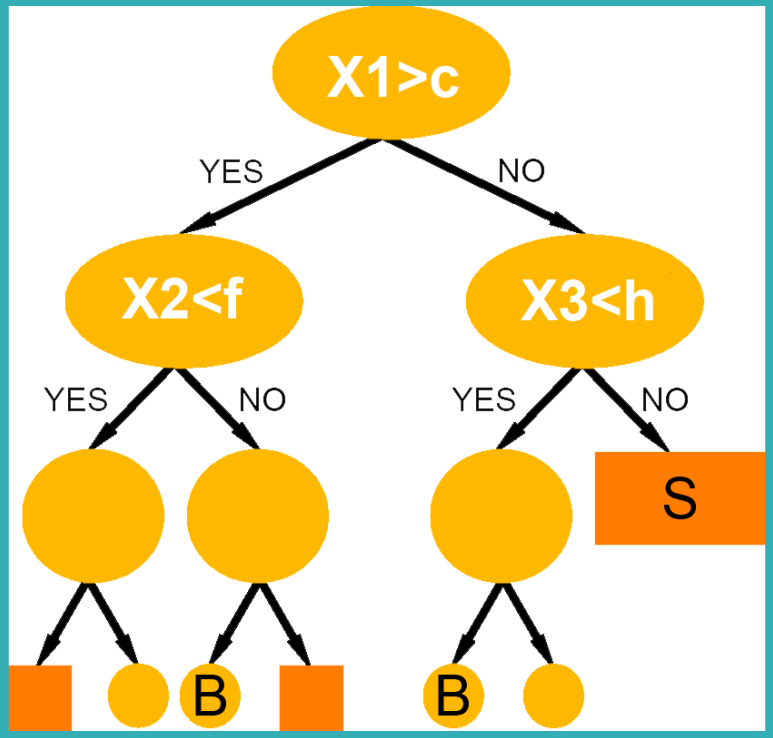
This says Parallelize me!


20 Folds
1,000,000 Events
Entries 6
Mean 1.788
Std Dev 1.08
Time (min)
0
5
10
15
20
25
30
35
0
1
2
3
4
5
6
7
Number of Processors

Outlook

Parallelization of different methods like

BDT ⟶ Boosted Decision Tree



Thank you all very much
for your attention!