Parallelization of ROOT Machine Learning Methods

Pourya Vakilipourotakalou

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!
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.
• 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

- $X \rightarrow$ vector of Kinematic Variables
- $Y \rightarrow$ output: Higgs (Signal) or Background $\rightarrow$ 0 or 1
- Proposing a function like $F(X)$
ROOT is a modular scientific software framework mainly written in C++ and integrated with other languages such as Python and R. It provides functionalities for big data processing and statistical analysis.
TMVA

ROOT, Machine Learning $\rightarrow$ 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

<table>
<thead>
<tr>
<th>Complete dataset</th>
<th>Training dataset</th>
<th>Test dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>fold 1</td>
<td>fold 2</td>
<td>fold 3</td>
</tr>
<tr>
<td>Test Set</td>
<td>Test Set</td>
<td>Test Set</td>
</tr>
</tbody>
</table>

K-fold cross validation (k=4)

- 1\(^{st}\) iteration → Calc. error
- 2\(^{nd}\) iteration → Calc. error
- 3\(^{rd}\) iteration → Calc. error
- 4\(^{th}\) iteration → Calc. error

Calculate avg. error
Cross Validation: PlotROC()

ROC → Receiver Operating Characteristic: a graphical plot that illustrates the performance of a classifier.
Parallelization

ROOT Classes for Parallelization

• ThreadPool $\rightarrow$ Multithreading
• TProcPool $\rightarrow$ Multiprocessing

• Multithreading $\rightarrow$ More difficult to implement: needs locking $\rightarrow$ no Global Variable

• Multiprocessing is easier but in some cases slower

This says Parallelize me!
Outlook

Parallelization of different methods like BDT $\rightarrow$ Boosted Decision Tree
Thank you all very much for your attention!