Name of the thesis

Konstantinos Papadimos

Contents

1 Theoritical Overview

1.1 Units in particle physics

Particle physics, study the properites of subatomic particles and their interactions. To describe such microscopic phenomena and quantities, an appropriate system of units must be adopted, for if the SI system of units was to be used, one would have to deal with large exponents. More over, it is more practical to utilize a system that is based on the typical length and time scales found in particle physics[modern particle phyics].

The foundamental constants of the special theory of relativity, as well as quantum mechanics are $\hbar$, c, and GeV and they can be used as a basis to form a system of units, the Natural Units. Natural units can be further simplified by chosing:

$$c = \hbar = 1$$

The table below sumarizes the relationship between natural units and S.I

Relationship between natural units and S.I		
Quanity	Natural units($\hbar = c = 1$)	S.I
Energy	GeV	Kgm^2s^{-2}
Momentum	GeV	Kgm^2s^{-2}
Mass	GeV	Kg
Time	GeV^{-1}	s
Length	GeV^{-1}	m

Table 1: Some basic quantites in S.I and in Natural units .

1.2   Special Relativity

1.2.1   Four-Vectors - Lorentz transformations

Let S and S' be two inertial reference systems, with S' moving at (a relativistic) velocity u relative to S. The coordinates are chosen such that the motion is along the x-axis in both reference systems. The clocks of both S and S' are synchronized so that when $x = x' = 0$, $t = t' = 0$. For an event with coordinates (x, y, z, t) in S, the coordinates in S' are given by the Lorentz transformations:

$$
\begin{aligned}
(x')^0 &= \gamma(x^0 - \beta x^1) \\
(x')^1 &= \gamma(x^1 - \beta x^0) \\
(x')^2 &= x^2 \\
(x')^3 &= x^3
\end{aligned}
\tag{1}
$$

where $\beta \equiv \frac{u}{c}$, $x^0 \equiv ct$, $x^1 \equiv x$, $x^2 \equiv y$, $x^3 \equiv z$

The elements x^i, i = 0, 1, 2, 3 define the position four vector. Mathematicaly four vectors are 4 dimentional, rank 1 tensors that transform according to lorentz transformations

With the introduction of four vectors, using Einstein's summation convention, lorentz transformations can be written as:

$$
(x')^i = \Lambda^i_j x^j
\tag{2}
$$

Where Λ, is the Lorentz transformation matrix, a rank 2 tensor:

$$
\Lambda = \begin{pmatrix}
\gamma & -\gamma\beta & 0 & 0 \\
-\gamma\beta & \gamma & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\tag{3}
$$

The following quantity does not change under lorentz transformation:

$$
I^2 = -(x^0)^2 + (x^1)^2 + (x^2)^2 + (x^3)^2 = -(x'^0)^2 + (x'^1)^2 + (x'^2)^2 + (x'^3)^2
\tag{4}
$$

or written in a more compact form:

$$
I = g_{\mu\nu} x^\mu x^\nu = x^\mu x_\mu
\tag{5}
$$

where $g_{\mu\nu}$ is the Minkowski (metric) tensor:

$$
g_{\mu\nu} = \begin{pmatrix}
-1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1
\end{pmatrix}
\tag{6}
$$

Such a quantity as I is called **invariant**

The introduction of four vectors and the metric tensor, yield the Minkowski Space Time where points need 4 coordinates to be fully described(1, time like and 3 space like) and the distance between them is beeing defined by the Minkowski tensor. The necessity of four vectors, in order to describe non scalar quantities(such as velocity and momentum) in minkowski space time, is therefore evident.

1.2.2   Energy and Momentum

According to the principle of relativity, the laws of physics must be the same in all inertial reference systems. Hence, if momentum is conserved in one inertial frame of reference, it must also be conserved in all others. It is evident that the momentum of a moving particle must be defined in an appropriate manner, to satisfy the principle of relativity [3]. The four momentum is therefore defined as:

$$
p^\mu = m\eta^\mu
\tag{7}
$$

Where $\eta^\mu = \frac{dx^\mu}{dt'}$, the four velocity of the particle.

The timelike component of four momentum, expresssed in natural units is $p^0 = \gamma m$. The 3 space like components, constitue the vector momentum, $\vec{p} = \gamma m \vec{\beta}$. The relativistic energy is defined as:

$$E = \gamma m = p^0 \tag{8}$$

Thus, the components of four momentum are:

$$p^\mu = (E, \vec{p}) \tag{9}$$

At this point we are able to calculate the invariant "interval" $p^\mu p_\mu$:

$$p^\mu p_\mu = E^2 - |\vec{p}|^2 = m^2 \tag{10}$$

What is invariant in the case of momentum four vector, is the particle's mass. This quantity is called **invariant mass** of a particle as all observers, in different frames of references, agree upon its value. It is the invariant mass of particles that we can(or we try to) measure, in CERN as well.

1.3   The standard model

1.4   Collisions and resonances

1.5   Maybe some elements of the theory behind our monte carlo simulations

2   Detector physics, the CMS detecor

3   Machine Learning for Classification

The main goal Machine Learning is set to achieve, is the development of algorythms equiped with the capabillity of learning from data automatically. In particular, an artificially intelligent system must have the ability to identify objects in its surroundings, as well as anticipate the actions of its environment, in order to make informed decisions. Due to this, machine learning techniques tend to be more oriented towards forecasting, rather than prediction.

3.1   Decision trees and Supervised Learning

The distinction between different particles, can be regarded as a classification problem where the target, is the prediction of a categorical output variable(i.e. lepton, boson), based on one or more input variables(i.e momenta components). Classification problems in machine learning can be solved with supervised learning. In such procedure, a training data set is being used for the development(training) of a model that is able to perform the classification task. The output of the model is then beeing tested and evaluated on previously unseen data.

Before presenting any specific method of solving classification problems It is important to present an overview of the key elements in supervised learning.

3.1.1   Supervised learning

Let us pose the following problem: Given a data set $D = (\vec{X}, \vec{y})$, where $\vec{X}$ is a matrix of the independet variables and $\vec{y}$ is a vector of dependent variables, we want to find a model $f(\vec{x}; \vec{\theta})$, that can predict an output from a set of input variables. Moreover, we want to be able to judge the performance of the model on a given data set. To do that we need to define a cost function $C(\vec{y}, f(\vec{X}; \vec{\theta}))$, such that the model will have to find the parameters θ that minimize the cost function.[5]

This the mathimatical pustulation of a supervised learning problem. I will now, in brief, discuss the role and interpretation of each of the 'ingredients' stated above.

- Model

The model, is a mathematical function $f : \vec{x} \rightarrow y$ of the parameters θ. Given a set of parameters, the output of the function, the prediction y_i, is derived from the input variables $\vec{x}$. The parameters are undefined. The task of the training is to estimate the set of parameters from the training data set. In a classification problem(something is of type a or it is not), it is possible to use the logistic transformation of the function output, to obtain the probability of the positive class.

- Cost function

The cost function, also known as an objective function, is represented by mathematical function and it measures how well a model fits the training data. The cost function is used to train the model by finding the best set of parameters θ that minimize the function. In machine learning, the objective function, usually consists of two parts: a training loss function (L) and a regularization term (Ω).

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{11}$$

The training loss function measures how predictive the model is with respect to the training data. A common choice of training loss function is the logistic loss, which is used for logistic regression(classification) and is given by

$$L(\theta) = \sum_i [y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i \ln(1 + e^{\hat{y}_i}))] \tag{12}$$

where y_i is the true label and $\hat{y}_i$ is the predicted label.

The regularization term, $\Omega(\theta)$, controls the complexity of the model, which helps to avoid overfitting. Overfitting occurs when a model is too complex and starts to extract local features from the training data. The model thus, looses its generalization power to new unseen data. Regularization helps to prevent overfitting by adding a penalty term to the cost function, which discourages the model from having too many parameters or too complex a structure.

The following figure gives an example of overfitting due to a very complex and very simple model.

3.1.2 Decision Trees

A decision tree is a flowchart-like tree structure, where each internal node represents a feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome.

Formally, a decision tree can be represented as a set of rules or conditions in the form of:

```
f(X)={condition1,condition2,..condition_n}
```

where each condition is a tuple of the form (feature, threshold, comparison operator) and the final outcome is represented by the leaf node. For example, consider the decision tree of figure 2 that classify fruits based on color, shape, size, and taste. Let X be the input X = {"red", "smal", "sour"}. Then f(X) = "grape" [2]

- Decision Tree Ensembles

The tree ensemble model consists of a set of classification and regression trees (CART). Let $\mathcal{F}$ be the set of all possible CART's and $f_k \in \mathcal{F}$, a function that represents a CART. The model in discussion then, can be written as:

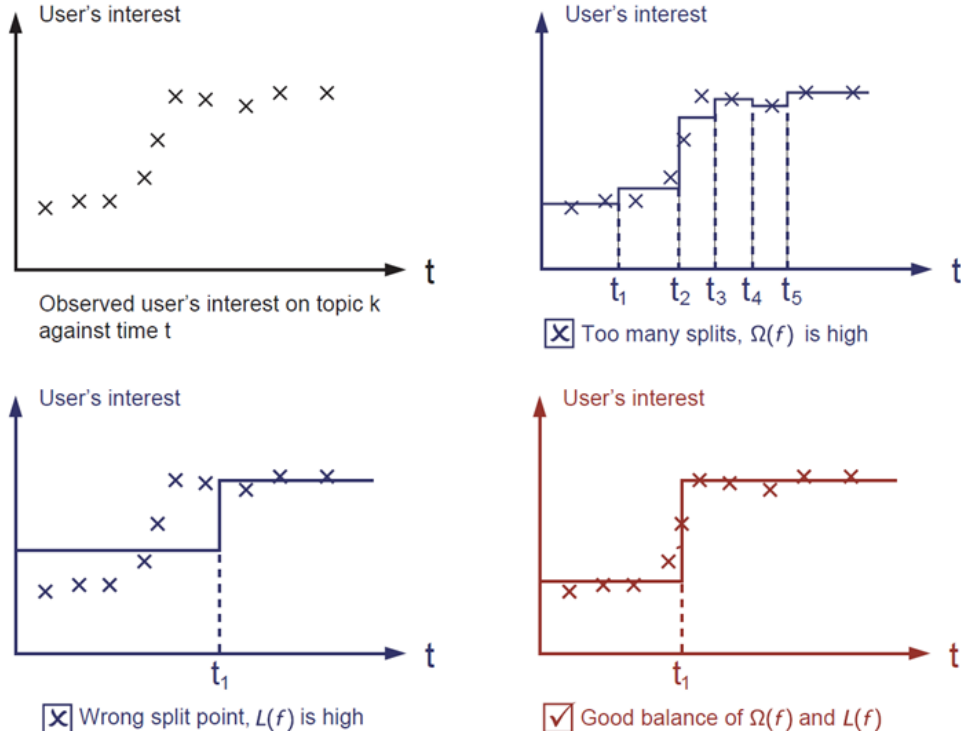$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \ f_k \in \mathcal{F} \tag{13}$$

4

Figure 1: Examples of over fitting and under fitting. The top right model, places too many cuts. Even though it succesfully describes the trend, the splits seem to correspond only on the specific data set, therefore it is overfitted. The bottom left model places too few and impresice cuts. The bottom right model seems to succesfully describe the trend while its simplicity infers that It has not sacrificed its generalization power.

If $\hat{y}_i$ represents the prediction of the tree, given an input variable x_i, the real label of x_i will be denoted as y_i . The objective function will be of the form:

$$obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{t} \omega(f_i) \tag{14}$$

where $\omega(f_i)$ is the complexity of a given tree and l is the loss function.

- Tree boosting

As stated earlier, the model is beeing trained, to learn those trees f_k that minimize the objective. The resulting model then, will be an ensemble of those functions f_k. The optimization of the objective, is a problem that cannot be solved with the traditional methods. Instead, the model is being iteratively trained in an additive manner.[1] let the prediction value at the t-th iteration be $\hat{y}_i^{(t)}$. In the next iteration(t+1), the chosen function f_{t+1}, is such that if added to the model, the resulting prediction $\hat{y}_i^{(t+1)}$ will minimize the cost function:

$$\hat{y}_i^{(0)} = 0$$
$$\hat{y}_i^{(1)} = \hat{y}_i^{(0)} + f_1(x_i)$$
$$\hat{y}_i^{(2)} = \hat{y}_i^{(1)} + f_2(x_i) \tag{15}$$
$$\ldots$$
$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) = \sum_{k=1}^{K} f_k(x_i)$$
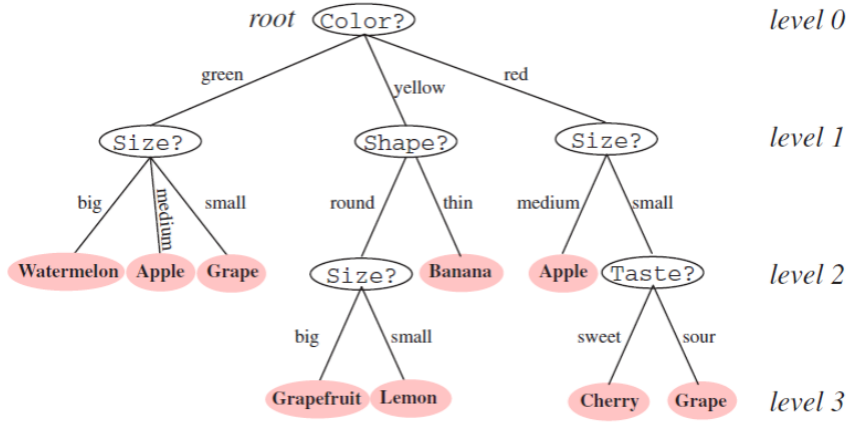
Figure 2: Example of a a decision tree that clasifies fruits

The objective at step t is:

$$obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t)}) + \sum_{i=1}^{t} \omega(f_i) = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \omega(f_i(t)) \qquad (16)$$

Taylor expanding the loss function $l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i))$, around f_t, up to the second order and neglecting terms, referring to previous rounds, the specific objective becomes:

$$\sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \omega(f_t) \qquad (17)$$

Where

$$
\begin{aligned}
g_i &= \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \\
h_i &= \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})
\end{aligned}
\qquad (18)
$$

This is the minimization goal for f_t . [4]

References

[1] Tianqi Chen and Carlos Guestrin. "XGBoost". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, Aug. 2016. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145%2F2939672.2939785.

[2] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* Wiley, 2006. ISBN: 9788126511167. URL: https://books.google.com.cy/books?id=NR-SzW2t7WYC.

[3] D. Griffiths. *Introduction to Elementary Particles.* Physics textbook. Wiley, 2008. ISBN: 9783527618477. URL: https://books.google.com.cy/books?id=Wb9DYrjcoKAC.

[4] *Introduction to Boosted Trees.* URL: https://xgboost.readthedocs.io/en/stable/tutorials/model.html (visited on 01/28/2022).

[5] Pankaj Mehta et al. "A high-bias, low-variance introduction to Machine Learning for physicists". In: *Physics Reports* 810 (May 2019), pp. 1–124. DOI: 10.1016/j.physrep.2019.03.001. URL: https://doi.org/10.1016%2Fj.physrep.2019.03.001.