Machine Learning for Higgs Boson Measurement

James Shinner
Level 4 Project, MSci Natural Sciences
Supervisor: Professor M. Spannowsky
Department of Physics, Durham University

April 20, 2018

Abstract

The aim of this project is to use machine learning methods to distinguish signal events from background events for two different decay modes of the Higgs boson. The investigated decay channels are $H \to ZZ^* \to \mu^+\mu^-e^+e^-$ ($2\mu2e$) and $H \to WW^* \to \mu^+\nu_\mu\mu^-\bar{\nu}_\mu$ ($2\mu2\nu$). The machine learning method utilised is boosted decision tree algorithms. Specifically, we implement the CART model of decision trees, and generate weak learning rules using decision stumps. These are combined to obtain a strong classifier using the ensemble algorithm known as AdaBoost. This algorithm is implemented from scratch in Python and trained on 20,000 pseudodata events. By testing different parameter combinations, we achieve the following best results - a classification error of 8.3% for the $2\mu2e$ channel, and a classification error of 21.65% for the $2\mu2\nu$ channel. The performance of our AdaBoost implementation with decision stumps is compared to the performance of the AdaBoost implementation in the open-source Python package scikit-learn. We find that our implementation is marginally beaten by scikit-learn on the $2\mu2e$ channel, but outperforms scikit-learn on the $2\mu2\nu$ channel.

Contents

1 Background and Theory

In this section, we explore the theory and background relevant to this project. It is split into two parts, each focusing on a different relevant aspect of this background.

In §1.1, we introduce the standard model and spontaneous symmetry breaking, explore Higgs phenomenology, and give an overview of collider physics.

In §1.2, we look at the machine learning algorithms and techniques that will be utilised in this project - decision trees and the AdaBoost algorithm.

1.1 The Higgs Boson & Collider Physics

The Higgs boson is a scalar particle predicted by the standard model of particle physics. Along with the Higgs field, it is named after Peter Higgs, who first postulated its existence.[1] We begin this section with a brief overview of the standard model and why this theory leads us to the prediction of a Higgs boson.

The theoretical predictions of the standard model are confirmed experimentally using particle collider experiments. We look at how such experiments were used to confirm the existence of the Higgs boson, and the statistical significance required to confirm a discovery. Then we look at how such experiments measure particle collisions, the experimental variables used, and the ways in which Higgs bosons are produced and decay at colliders.

1.1.1 Particles of the Standard Model

The standard model of particle physics is a mathematical description of the fundamental particles and interactions that make up the universe.[2] Formulated in the second half of the 20th century, it has proved to be consistent with many experimental predictions.

The particles of the standard model can be divided into *fermions* and *bosons*, with fermions having ½-integer spin and bosons having integer spin.

The fermions can be further divided into *quarks* and *leptons*, which make up atomic matter. Quarks are differentiated from leptons by possessing a colour charge, meaning that they experience interactions through the strong force in addition to electroweak interactions, whereas leptons only experience interactions of the latter kind. The quarks are divided in two; up, charm and top quarks have electric charge +⅔ whilst down, strange and bottom quarks have electric charge -⅓. The leptons are also split, with electrons, muons, and tauons having electric charge -1 and neutrinos being neutrally charged, thus only interacting via the weak force.

Both quarks and leptons are divided into generations, with particles of higher generations having a greater mass than the corresponding particles of lower generations, but undergoing identical interactions. Particles of higher generations are only found at high energies and decay into lower generations, which do not decay. All fermions also have a corresponding antiparticle.

The bosons are divided into *scalar bosons*, with spin 0, of which the only fundamental example is the Higgs boson, and *gauge bosons*, with spin 1. The gauge bosons are said to mediate the strong, weak and electromagnetic force interactions. The gluon is the force carrier for the strong force, the photon serves as the force carrier for the electromagnetic force and the Z and W bosons mediate the weak force.

1.1.2 Spontaneous Symmetry Breaking

Mathematically, the standard model is a renormalisable quantised gauge field theory.[3][4]. It is said to have $SU(3)_c \times SU(2)_L \times U(1)_Y$ symmetry. Here c denotes *colour*, L denotes *left*, and Y denotes *hypercharge*.

According to Noether's Theorem, each of these gauge symmetries leads to a conserved quantity.[3] Colour charge is the conserved quantity for $SU(3)_c$ symmetry, while the conserved quantity for $SU(2)_L$ symmetry is weak hypercharge and $U(1)_Y$ symmetry is associated with the conservation of electric charge.

Each of the symmetries are associated with a gauge field and a force. $SU(3)_c$ is associated with the strong force, and the force carrier is the gluon. The symmetry is preserved as the colour symmetry of quantum chromodynamics.[5]

The $SU(2) \times U(1)_Y$ symmetry represents a unified electroweak gauge field, unified by Glashow, Weinberg and Salam.[6][7][8]. However, unlike $SU(3)_c$, this symmetry is not preserved; it is said to be spontaneously broken.[9]

The spontaneous breaking of this $SU(2)_L \times U(1)_Y$ symmetry to $U(1)_{el}$ symmetry introduces the Higgs field, providing all fundamental particles with mass through the Higgs mechanism.[10] The gauge boson associated with the Higgs field is the Higgs boson.

For a detailed derivation of the Higgs mechanism and how it gives mass to the fundamental particles, see Appendix I.

1.1.3 The Discovery of the Higgs Boson

The Large Hadron Collider (LHC), located at CERN in Geneva, Switzerland, is the largest particle collider ever built, capable of reaching energies not previously accessible in experimental particle physics. It started its initial data collection run in 2010, at a centre of mass energy of 7 TeV.[11] The first evidence of a Higgs boson was seen in 2011, and a limit was imposed on the particle's mass, constraining it to below 127 GeV. In 2012, the centre of mass energy at the LHC was increased to 8 TeV. CERN confirmed the discovery of the Higgs boson with a mass of around 125 GeV in July 2012, via both the ATLAS and CMS experiments at the Large Hadron Collider.[12][13] Since then, the LHC has continued to collect data, with Run 2, begun in 2015, reaching a centre of mass energy of 13 TeV.[14]

1.1.4 Statistical Significance of Discovery

The statistical significance required for a discovery result to be announced in particle physics is known as 5σ,[15] corresponding to 5 standard deviations from the mean of a dataset.[16] This corresponds to a p-value of 3×10^{-7}. This is the probability that the collected data would be at least as extreme as it is observed to be, if a standard model

Higgs-like boson does not exist. This corresponds to an approximately 1 in 3.5 million chance.

The high level of confidence required to announce a discovery motivates the refinement of data analysis techniques, in order to ensure the data collected from the LHC is used efficiently, ensuring maximum statistical significance can be obtained from the data.

1.1.5 The Experimental Setup of the LHC

The LHC is located inside a circular underground tunnel with a 26.7km circumference. Strong electromagnets are used to accelerate beams of protons around the tunnel, until they reach the required energies, and then they are collided at one of the experiment's detectors. The products of the collision are then detected and analysed by these detectors.

The ATLAS and CMS detector experiments both reconstruct muons, electrons, photons and hadronic jets produced in proton-proton collisions.[17][18] The CMS experiment is also used to measure the missing transverse momentum carried by weakly interacting particles. The ATLAS detector uses a toroidal magnet system as its muon spectrometer, while the main feature of the CMS detector is a superconducting solenoid.

1.1.6 Experimental Variables and Observables

With the commencement of Run 2, the centre of mass energy of the proton beams reached 13 TeV. The beam intensity can be quantified by the *instantaneous luminosity*, $\mathcal{L}$, measured in $cm^{-2}s^{-1}$. Multiplying this by the cross section of a process, σ, gives the rate of interactions, $\dot{N}$. The *integrated luminosity* (instantaneous luminosity over time), $\mathcal{L}_{int}$, measured in fb^{-1}, quantifies the amount of data collected over a period of time. In 2017, the LHC reached a maximum instantaneous luminosity of $\mathcal{L} = 2.06 \times 10^{34} cm^{-2}s^{-1}$ and the combined integrated luminosity for ATLAS and CMS reached $\mathcal{L}_{int} = 50 fb^{-1}$.[19] This is equivalent to 5×10^{15} collisions over the course of the year.

Proton collisions at the LHC can lead to extremely complicated processes involving many different kinds of particle. The final states of these processes are measured by the detector experiments. These detectors record the *4-momentum* of the particles in these observed states, denoted $p = (E, p_x, p_y, p_z)$. E denotes the energy of the particle, while p_x, p_y, p_z denote the particle's momentum in each of the spatial directions. The z-axis is defined to be parallel to the proton beam, so p_z is the momentum along the beam axis and the *transverse momentum* $p_T = \sqrt{p_x^2 + p_y^2}$ is the momentum perpendicular to the beam axis. The detectors also measure the polar angle θ, and azimuthal angle ϕ, of the final state particle from this same beam axis.[20] A visualisation of the situation can be seen in Fig. 1.1.

From these measurements, we can also construct several other useful quantities. The *invariant mass*, m of the particle is given by:

$$\begin{aligned} m^2 = p^2 = E^2 - p_x^2 - p_y^2 - p_z^2 \\ m = \sqrt{E^2 - p_x^2 - p_y^2 - p_z^2} \end{aligned} \tag{1.1}$$

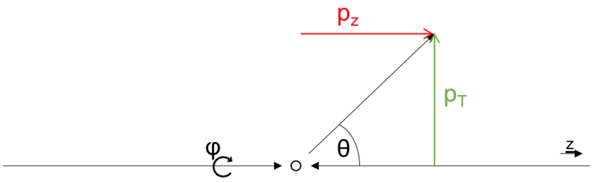


Figure 1.1: A visualisation of a proton-proton collision situation showing p_z, p_T, θ and ϕ, where the z-axis is along the proton beam axis.

The *rapidity*, y is given by:

$$y = \frac{1}{2}\ln(\frac{E+p_z}{E-p_z}) \tag{1.2}$$

This is a useful quantity as the difference in rapidity between two particles, Δy, is a Lorentz invariant quantity. Differences in the angular separation between particles can not be expressed using differences in the polar angle θ as this is not Lorentz invariant, and different particles will have different velocities along the beam axis. We construct a Lorentz invariant measure of the angular separation from the rapidity and azimuthal angle ϕ (measured in the transverse plane, so $\Delta\phi$ is Lorentz invariant). This is known as the ΔR separation:

$$\Delta R = \sqrt{\Delta y^2 + \Delta\phi^2} \tag{1.3}$$

1.1.7 Higgs Boson Interactions

The interaction of a particle with the Higgs boson depends on its mass, so Higgs bosons are mostly produced in association with heavy quarks and the massive W and Z gauge bosons. The four dominant production modes are, in order of largest cross section: (a) gluon fusion via a top quark loop, (b) fusion of W/Z bosons radiated from quarks, (c) associated production with a W/Z boson, and associated production with a $t\bar{t}$ quark-antiquark pair.[10] These four production modes are shown in Fig. 1.2

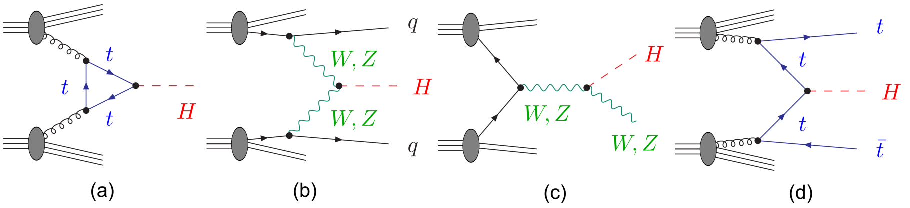


Figure 1.2: The dominant production modes for the Higgs boson at the LHC.

Higgs boson production channels have a very small cross section when compared to the main background processes. In the data collected up until June 2012 that was used to announce the discovery, around 400,000 expected Higgs bosons were produced, out of a total of 10^{15} proton-proton collisions. Three main decay modes were examined to find evidence for the existence of the particle: the decay to four leptons via a real and virtual

Z boson, the decay to two leptons and two neutrinos via a real and virtual W boson, and the decay to two photons.[21] The equations for these processes are:

$$H \to ZZ^* \to 4l, \quad H \to WW^* \to 2l2\nu, \quad H \to \gamma\gamma. \tag{1.4}$$

In this project, we look at cases of the first two decay channels, specifically the decay to muons and electrons, and the decay to muons and muon neutrinos:

$$H \to ZZ^* \to \mu^+\mu^-e^+e^-, \quad H \to WW^* \to \mu^+\nu_\mu\mu^-\bar{\nu}_\mu. \tag{1.5}$$

1.1.8 $H \to ZZ^* \to \mu^+\mu^-e^+e^-$

The Feynman diagram for the decay of the Higgs boson to a muon, electron, antimuon and positron via a real and virtual Z boson can be seen in Fig. 1.3.

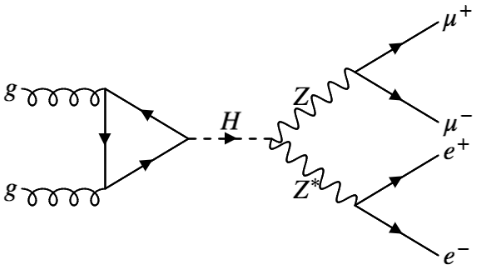


Figure 1.3: The $H \to ZZ^* \to \mu^+\mu^-e^+e^-$ decay.

By combining the 4-momentum of the final state leptons, we can reconstruct combined invariant masses, which can be used to discriminate the signal from background processes. If μ^+, μ^-, e^+, e^- have momentum p_1, p_2, p_3, p_4 respectively, we can construct $m_{2\mu}$, m_{2e} and m_{4l} as follows:

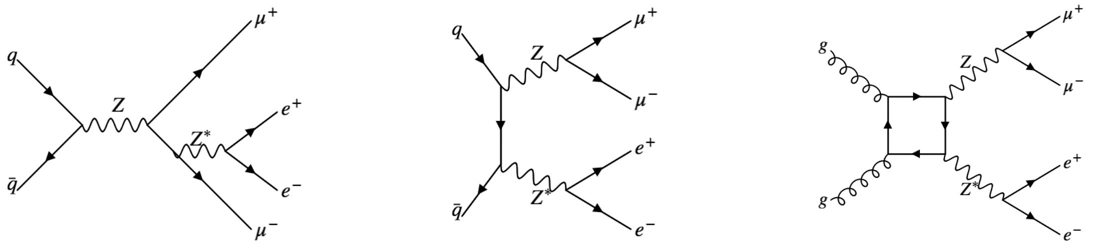
$$m_{2\mu}^2 = (p_1 + p_2)^2 \quad m_{2e}^2 = (p_3 + p_4)^2 \quad m_{4l}^2 = (p_1 + p_2 + p_3 + p_4)^2 \tag{1.6}$$

For the signal process, we see that $m_{2\mu}$ and m_{2e} correspond to the masses of the on shell and off shell Z boson and m_{4l} corresponds to the mass of the Higgs boson.

Several background processes produce similar 4-lepton final states. The dominant background processes are the production of Z boson pairs without an intermediate Higgs boson, from a $q\bar{q}$ or gg initial state.[22] These processes are shown in Fig. 1.4. The contribution from the $q\bar{q}$ processes is greater than the contribution from gg (Fig. 1.4c). Of the two $q\bar{q}$ channels, the s-channel process (Fig. 1.4a) dominates at the Z-boson resonance, while the t-channel process (Fig. 1.4b) dominates at higher values of m_{4l}.

1.1.9 $H \to WW^* \to \mu^+\nu_\mu\mu^-\bar{\nu}_\mu$

The Feynman diagram for the decay of the Higgs boson into a muon, antimuon, muon neutrino and muon antineutrino via a real and virtual W boson can be seen in Fig. 1.5.



(a) s-channel $q\bar{q}$ process **(b)** t-channel $q\bar{q}$ process **(c)** gg box process

Figure 1.4: The dominant background processes of $H \to 2\mu 2e$.

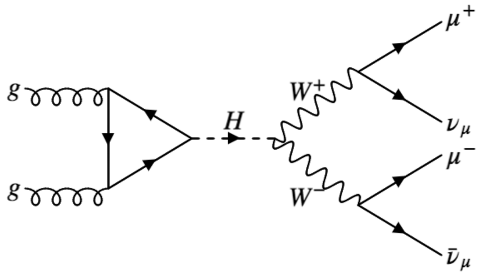
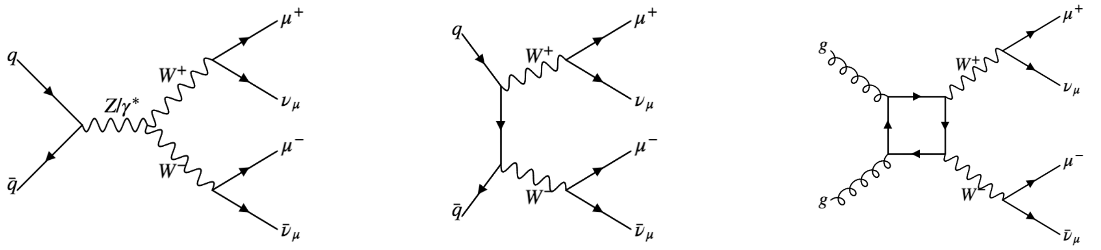


Figure 1.5: The $H \to WW^* \to \mu^+\nu_\mu\mu^-\bar{\nu}_\mu$ decay.

The neutrinos escape the detector, so the 4-momentum of the neutrinos is not known, nor can we construct the combined invariant masses of the Higgs and W bosons as in §1.1.8.

The dominant background processes are the non-resonant direct production of W boson pairs from s-channel and t-channel $q\bar{q}$ annihilation (Fig. 1.6a & Fig. 1.6b),[23] along with gg fusion (Fig. 1.6c).[24]



(a) s-channel $q\bar{q}$ process **(b)** t-channel $q\bar{q}$ process **(c)** gg box process

Figure 1.6: The dominant background processes of $H \to 2\mu 2\nu$.

1.2 Machine Learning

Machine learning involves using algorithms to discover patterns in a dataset. This section begins with a general exploration of the machine learning problem. We then introduce decision tree learning and the AdaBoost algorithm - the methods implemented for this project.

1.2.1 Theory of Machine Learning

Machine learning can be broken down into supervised learning, where the algorithm is trained to make predictions using existing labelled data, and unsupervised learning, where the algorithm must find patterns in unlabelled data. The method used in this project is an example of supervised learning.

Supervised learning can be further broken down into two classes of problem: classification and regression. Regression problems involve making a prediction for a value on a continuous range. Classification problems involve making a prediction from a discrete set of possible values, such as answering a yes/no question. For this project, we have a classification problem with two possible options - signal and background.

A 'well-posed' machine learning problem can be defined as follows:[25]

"A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

For this project, the task T is classifying events as signal or background, the performance measure P is the percentage of events correctly classified, and the experience E is the set of training data used to train the algorithm.

This general setting of the supervised machine learning problem can be expanded by introducing the concept of a hypothesis function, usually written $h(\mathbf{x})$, where $\mathbf{x}$ is the input data. This hypothesis function acts as the selection rule used to make a prediction about a data point. Using the training data, a given algorithm will tweak the parameters of this hypothesis function. The explicit mathematical form of $h(\mathbf{x})$ and the manner in which the parameters are changed depends on the algorithm in question.

There are many different machine learning algorithms, with the suitability of different approaches depending on the intricacies of the problem being solved. In this project, we will be using a boosted decision trees method.

1.2.2 Decision Trees

Decision tree learning is a popular method for many machine learning problems. Primarily used for classification tasks, some variations of decision trees are also suited for regression problems. Decision trees are easily visualised as flowcharts, as can be seen in Fig. 1.7.[26]

Decision trees make predictions using a series of nodes. At each node, a specific attribute of the dataset is tested, and the data is sorted into branches based on the possible values of this attribute. Each branch leads to a new node, which can repeat the process of splitting or make a prediction. The nodes at which predictions are returned are known as *leaf nodes*, whilst the first node of a decision tree is known as the *root node*.

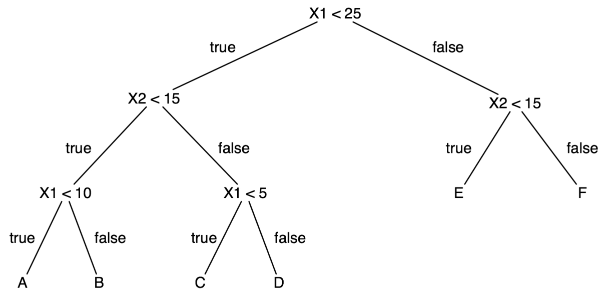


Figure 1.7: An example of a decision tree of depth 3.

The highest number of nodes used by the tree to make its predictions is known as the *depth* of the tree. Decision trees with a depth of 1 (i.e a single root node branching straight to leaf nodes) are known as *decision stumps.*

For a classification problem, the predictions at the leaf nodes will be one of the possible classification labels (1 or -1, yes or no, signal or background, etc.), and for a regression problem, the prediction is a value from a continuous range.

The basic decision tree algorithm known as ID3 (Iterative Dichotomiser 3) was introduced by Quinlan.[27] It can only handle discretely valued attributes, making it unsuitable for us as we have mostly continuous valued attributes.

A successor to the ID3 algorithm, the CART (classification and regression trees) algorithm was introduced by Breiman et al.[28] The algorithm measures how good a given split is using the splitting criterion of *Gini impurity*,[29] given by:

$$H_{Gini}(S) = 1 - \sum_{x \in X} p(x)^2 \tag{1.7}$$

For a perfectly classified set, we have $H_{Gini}(S) = 0$, whilst a set with a ${}^{50}/_{50}$ split between two classes has $H_{Gini}(S) = 0.5$. The information gain from splitting a set S on attribute A is defined as:

$$IG(A, S) = H_{gini}(S) - \sum_{t \in T} p(t) H_{gini}(t), \tag{1.8}$$

where $H_{gini}(S)$ is the gini impurity as defined above, T is the set of subsets created from splitting set S on attribute A, and $p(t)$ is the proportion of elements in subset t. The algorithm selects the split with the maximum information gain to be used at each node.

The CART algorithm improves on the ID3 algorithm by handling continuous as well as discrete values. It does this by considering a binary split on the attribute, sorting the data into two bins of below and above this split. The split point is determined by considering a number of split points on the data and choosing the split with the maximum information gain. The more splits used, the more computationally intensive the algorithm is, although the more likely it is to find the optimum splitting point.

Decision trees grown to fully classify every point in a dataset will reach a high depth, and often are severely affected by the problem of *overfitting.* This is when the learning hypothesis fits too closely to the training data, and picks up on noise and random fluctuations in the data, and does not generalise well to new unseen data.

To avoid this problem, trees can be stopped upon reaching a preset depth, using a method known as pruning to remove sections of the tree that have little predictive use, or combining multiple trees with a small depth using an ensemble method.

Ensemble methods combine multiple machine learning classifiers to obtain a hypothesis capable of making better predictions. They are often seen to reduce the overfitting tendencies of the base learning method, while improving the prediction accuracy. This project makes use of a boosting technique, namely the AdaBoost algorithm, where multiple weak learners (often performing little better than a random guess) are combined to produce a strong learner.[30]

1.2.3 The AdaBoost Algorithm

The AdaBoost algorithm (short for Adaptive Boosting) was introduced by Freund and Schapire.[31][32][33] It works by iteratively training weak learners on a dataset and combining them to create the final ensemble hypothesis. With each iteration, the events in the training sample are reweighted so that events that are misclassified are given more weight in the next iteration, while correctly classified events have a reduced weight, causing the algorithm to focus on the events for which it produces an incorrect prediction.

The algorithm for discrete AdaBoost in a binary classification setting is set out in Fig. 1.8.

```
Initialise the event weights to $w_i = \frac{1}{n}$, where $i = 1, ...., n$.
Training events $\mathbf{x} = (x_1, ..., x_n)$, training labels $\mathbf{y} = (y_1, ..., y_n)$ with $y_i \in \{-1, 1\}$, set of weak classifiers $h(\mathbf{x})$

For $t = 1, ..., T$, with T the number of boosting iterations:
    · Select weak classifier $h_t(\mathbf{x})$ that minimises misclassified weighted error $\epsilon_t$
    · Set $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$
    · Add weak classifier to ensemble hypothesis: $H_t = H_{t-1}(\mathbf{x}) + \alpha_t h_t(\mathbf{x})$
    · Update weights: $w_{i,t+1} = w_{i,t}e^{-y_i \alpha_t h_t(x_i)}$
    · Renormalise weights so that $\sum_i w_{i,t} = 1$

Output final hypothesis $H_T(\mathbf{x}) = \sum_t \alpha_t h_t(\mathbf{x})$. Take $sgn(H_T(\mathbf{x}))$ to be predicted class label.
```

Figure 1.8: The AdaBoost algorithm.

The final hypothesis $H_T(\mathbf{x})$ is a real-valued function. The output values depend on the values of alpha produced in the training process. The $sgn(H_T(\mathbf{x}))$ function is used to predict the class label for a given event - so if $H_T(x_i)$ is positive, the algorithm will predict a label of 1, while a negative value of $H_T(x_i)$ means the algorithm will predict a label of -1. However, more information is given by looking at the output of $H_T(\mathbf{x})$ before applying the *sgn* function. A larger positive value of $H_T(x_i)$ implies a more confident prediction of a class label of 1, while a larger negative value implies a more confident prediction of -1. A value of $H_T(x_i)$ closer to 0 equates to a less confident prediction.

The algorithm only holds if the base learners included in the ensemble have classification error < 0.5. If a weak learner is added to the ensemble with error equal to 0.5, the events will not be reweighted and further rounds of boosting will not be possible. This requirement for the error of all the weak classifiers to be strictly less than 0.5 is formally defined as the *weak learning condition*[34]:

$$\epsilon_t \leq \frac{1}{2} - \gamma_t,\ for\ some\ \gamma_t > 0 \tag{1.9}$$

The AdaBoost algorithm greatly improves on the performance of the base weak classifier that it utilises. In fact, it can be shown that there is an upper bound on the training error on a dataset.[35] After T rounds of boosting, the upper bound is:

$$training\ error \leq (\sqrt{1 - 4\gamma^2})^T \leq e^{-2\gamma^2 T}, \tag{1.10}$$

where γ is the minimum of γ_t. As we saw in Eq. 1.9, γ_t is the amount by which the error of a weak classifier is better than a random guess. The consequence of this upper bound is that we see that for any value of γ, the training error of AdaBoost decreases exponentially in relation to the number of rounds of boosting.

Whilst no such bound can be proved for the error on a test dataset, the performance increase of using AdaBoost is also notable when compared to the performance achieved by a base weak learner. The predictions of the algorithm also seems to be more resistant to overfitting when compared to decision trees.[34]

This project will involve an implementation of the AdaBoost algorithm as in Fig. 1.8, using decision trees of depth 1 (decision stumps) as the base weak learner.

2 Method

In this section, we give a detailed breakdown of the methodologies used to complete this project and the investigations that were carried out.

An overview of the dataset and how the pseudodata was generated is given in §2.1.

The algorithm is first trained using a base parameter set, and then using additional parameters to see how performance differs. In §2.2, we look at these parameters and how they were constructed from the data, along with a comparison of how these parameters are distributed for signal and background events.

The procedure of implementing the algorithm is discussed in §2.3, along with how the steps taken to ensure the weak learning condition was satisfied.

In §2.4, we introduce the `sklearn` package for Python. The performance of the AdaBoost implementation from this package will be compared to the implementation produced for this project.

The section is concluded in §2.5 with a look at how the performance of the learning algorithm will be quantified and the metrics used to determine the success of a learnt hypothesis function.

2.1 The Dataset

The dataset that was provided for this project is pseudodata - created by a simulation to replicate events that would be detected at a particle collider. The pseudodata was generated using a Monte Carlo simulation method by the event generator software MadGraph5_aMC@NLO.[36]

The pseudodata is formatted in Les Houches Event File, a standard format agreed upon for use in particle physics to standardise and ease the exchange of information between event generators and analysis software.[37] These events are read into a Python environment using the `pylhe` package and the data is then stored and manipulated as dataframas using the `pandas` package.

The pseudodata contains events for both the $H \to \mu^+\mu^-e^+e^-$ channel, as described in §1.1.8, and the $H \to \mu^+\nu_\mu\mu^-\bar{\nu}_\mu$ channel, as described in §1.1.9. These will henceforth be referred to as the $2\mu2e$ and $2\mu2\nu$ channels. The data for each channel consists of 20,000 labelled events, split equally between 10,000 signal events and 10,000 background events.

The data for each channel is split into two equal portions to create a training dataset and a validation dataset. The labelled training dataset is used by the algorithm to produce the hypothesis function. The algorithm then makes predictions on the unlabelled validation dataset, which are compared to the known event labels to measure the performance of the algorithm. Each of these datasets contains equal amounts of signal and background events. So for each channel, there is a training dataset of 10,000 events (5,000 signal and 5,000 background) and a validation dataset of 10,000 events (5,000 signal and 5,000 background).

For each event in the data, there are four final state particles. The data contains the particle ID of each particle according to the Monte Carlo particle numbering scheme,[38] along with the components of the 4-momenta of the particles - E, p_x, p_y, p_z. As discussed in §1.1.9, the 4-momentum of neutrinos cannot be measured by the detectors at a collider, so we cannot make use of the pseudodata for these particles when training our algorithm on the $2\mu2\nu$ channel.

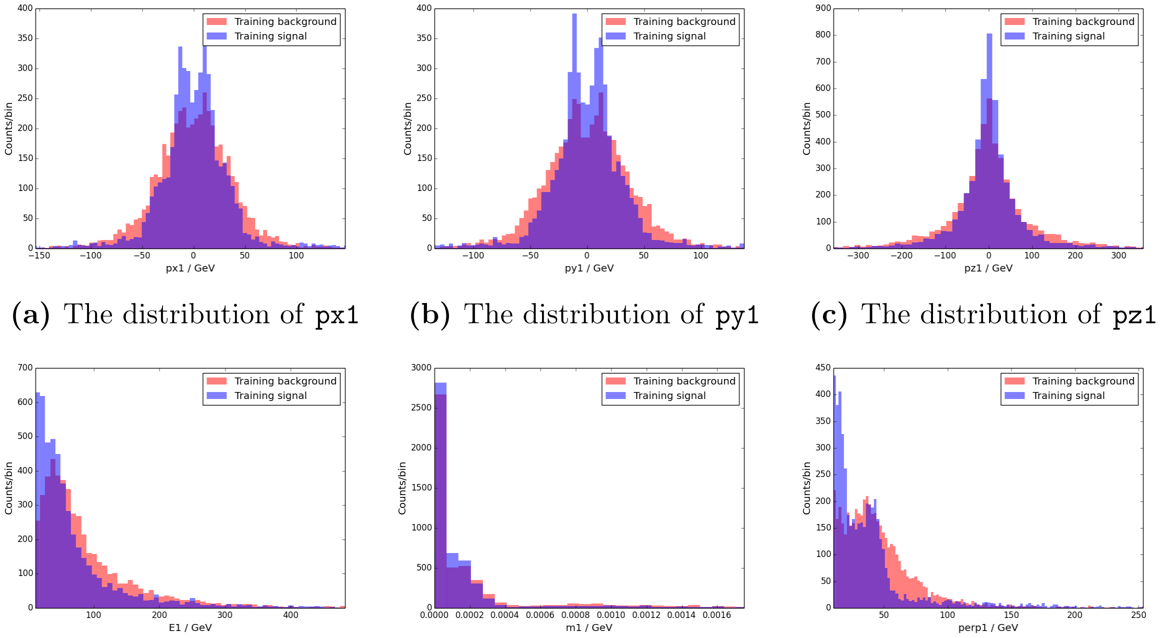
2.2 Training Parameters

We now describe the different parameters used to train the algorithm. The base parameter set was used to test the initial performance of the algorithm, and then additional parameters were added in an attempt to improve this performance.

2.2.1 Base Parameter Set

The base parameter set consists of the components of the 4-momentum for each particle in an event, along with their invariant masses and transverse momenta, constructed from the 4-momentum as discussed in §1.1.6.

This means the base parameter set for the $2\mu 2e$ channel was `px1,py1,pz1,E1,m1,perp1, px2, py2, pz2, E2, m2, perp2, px3, py3, pz3, E3, m3, perp3, px4, py4, pz4, E4, m4, perp4`, with particle 1 being μ^+, particle 2 being μ^-, particle 3 being e^-, and particle 4 being e^+. These parameters were plotted for the training dataset to compare their distributions for signal and background. A selection of these plots is shown in Fig. 2.1, showing the parameters `px1`, `py1`, `pz1`, `E1`, `m1`, and `perp1` for particle 1 (μ^+). Similar distributions are seen for the parameters for the other three particles. From a first look, we see that there's not a lot of difference on any of these parameters between the signal and background events. However, there are some small differences on all parameters, with some having a better separation than others. For example, the separation in `perp1` (Fig. 2.1f) appears to be more pronounced than the separation in `m1` (Fig. 2.1e) and we expect a split on this parameter to be more likely to be used by our machine learning algorithm.

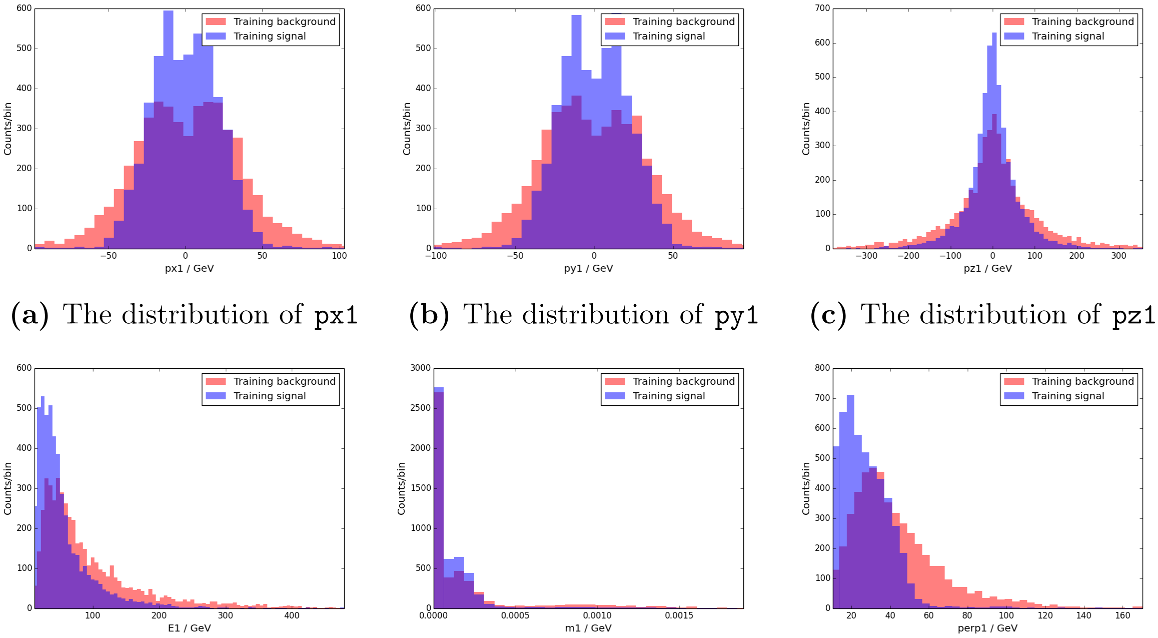


(a) The distribution of `px1` **(b)** The distribution of `py1` **(c)** The distribution of `pz1`

(d) The distribution of `E1` **(e)** The distribution of `m1` **(f)** The distribution of `perp1`

Figure 2.1: Plots of the base parameters for particle 1 (μ^+) on the training dataset of the $2\mu 2e$ channel, showing the distribution of the parameter values for the signal and background events.

For the $2\mu 2\nu$ channel, the base parameter set was `px1,py1,pz1,E1,m1,perp1, px2, py2, pz2, E2, m2, perp2`, with particle 1 again being μ^+, and particle 2 being μ^-. The parameters are only for these particles due to the missing information for the neutrinos. Similarly to above, the parameters `px1`, `py1`, `pz1`, `E1`, `m1`, and `perp1` for particle 1 (μ^+) are shown in Fig. 2.2. Again we see that the distributions are very similar for both the particles in the channel for which we can reconstruct the data. We also see that there is no major separation between signal and background for any of the features. However, by comparison with Fig. 2.1, we see that the separation between signal and background is slightly more pronounced in all features for this channel than for the $2\mu 2e$ channel.



(a) The distribution of `px1`

(b) The distribution of `py1`

(c) The distribution of `pz1`

(d) The distribution of `E1`

(e) The distribution of `m1`

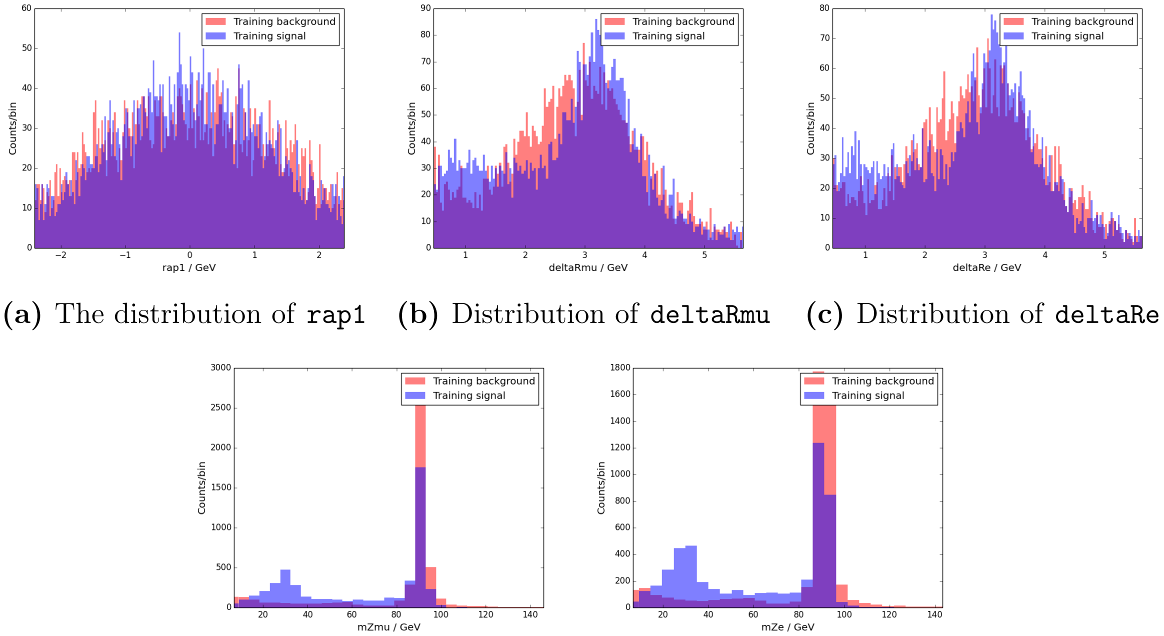
(f) The distribution of `perp1`

Figure 2.2: Plots of the parameters for particle 1 (μ^+) on the training dataset of the $2\mu 2\nu$ channel, showing the distribution of the parameter values for the signal and background events.

2.2.2 Additional Parameters

In addition to training the algorithm on the parameters described above, additional parameters were also constructed to see if performance of the algorithm could be improved. These additional parameters were the rapidity and ΔR separation as discussed in §1.1.6 and the reconstructed masses as discussed in §1.1.8.

For the $2\mu 2e$ channel, the rapidity is constructed for all four final state particles, the ΔR separation is constructed for each generation of leptons (so muon/antimuon separation and electron/positron separation), and the reconstructed masses used are $m_{2\mu}$, m_{2e}, and m_{4l} - these are expected to reconstruct to the masses of the Z bosons and the Higgs. This means the additional parameters added to the base parameter set were `rap1`, `rap2`, `rap3`, `rap4`, `deltaRmu`, `deltaRe`, `m2mu`, `m2e`, `m4l`. A representative set of these parameters is plotted in Fig. 2.3. All the `rap` parameters were found to have very similar distributions, so only `rap1` is plotted. From Fig. 2.3a, Fig. 2.3b and Fig. 2.3c, we see that again there is not a large separation between signal and background for either the rapidity or ΔR features. Fig. 2.3d and Fig. 2.3e show a more promising separation in the reconstructed mass features.



(a) The distribution of `rap1` **(b)** Distribution of `deltaRmu` **(c)** Distribution of `deltaRe`

(d) The distribution of `m2mu` **(e)** The distribution of `m2e`

Figure 2.3: Plots of the additional parameters implemented for the $2\mu 2e$ channel.

The distribution of `m4l` is plotted separately in Fig. 2.4. This shows a sharp spike at 125 GeV for the signal events, corresponding to the mass of the Higgs boson. This feature appears to be a very strong discriminator between signal and background. As we expect it to be such a strong discriminator, it is separated from the rest of the additional parameters and its effects on the learning process are investigated on its own.

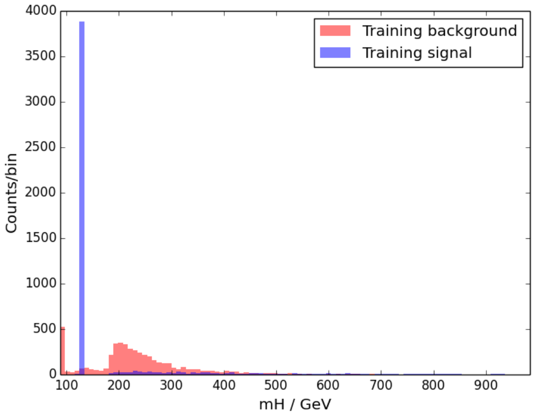
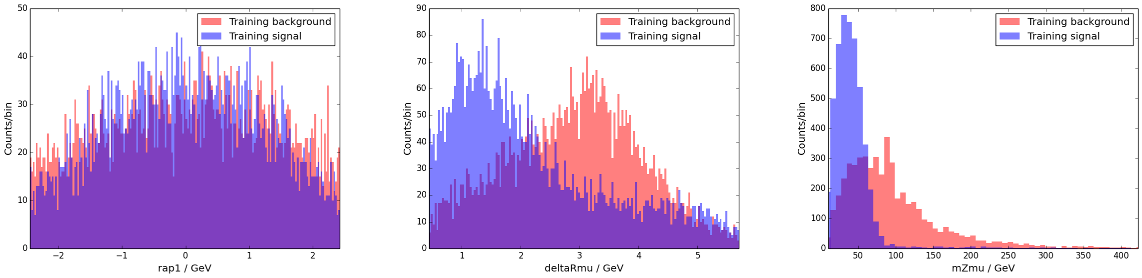


Figure 2.4: The distribution of `m4l` for the $2\mu 2e$ channel.

For the $2\mu 2\nu$ channel, we can only construct the rapidity for the μ^+ and μ^-, and the ΔR separation between these two particles. We also reconstruct the invariant mass $m_{2\mu}$, although this does not have the same physical significance as the reconstructed masses in the $2\mu 2e$ channel, as the muons are not produced from the same W boson. The additional parameters added to the base set were thus `rap1`, `rap2`, `deltaRmu`, `m2mu`. These features are shown in Fig. 2.5. Once again, `rap1` and `rap2` were found to be very similar, so only `rap1` is shown. Fig. 2.5a shows no significant separation for the rapidity, but Fig. 2.5b and Fig. 2.5c show a better separation between signal and background for ΔR separation and $m_{2\mu}$.



(a) Distribution of `rap1` **(b)** Distribution of `deltaRmu` **(c)** Distribution of `m2mu`

Figure 2.5: Plots of the additional parameters implemented for the $2\mu 2\nu$ channel.

The algorithm was trained using all the additional parameters (except `m4l` and `m2mu`), and performance was compared with the base parameter set. The effects of adding the parameters `m4l` on the $2\mu 2e$ channel and `m2mu` on the $2\mu 2\nu$ channel were tested separately as these parameters were seen to be particularly strong indicators of signal/background.

2.3 Implementation of the Algorithm

The first step in implementing the algorithm was to create the decision trees that would be used as the base weak learners. The CART algorithm was used, as discussed in §1.2.2, with the tree depth restricted to 1 to obtain decision stumps. The set of weak classifiers available to the boosting algorithm was a set of such decision stumps, with stumps splitting at various points on each parameter. The number of splits was varied from 10 splits on each parameter up to 10,000 splits, i.e. every possible split on the training dataset of 10,000 events.

The AdaBoost algorithm then picks the best of these decision trees at each algorithm, adds it to the ensemble and reweights the events to focus the next iteration on incorrectly classified events, via the process laid out in §1.2.3.

To begin with, a slightly different version of the AdaBoost algorithm was used than that which is presented in Fig. 1.8. Rather than picking the weak learner with the lowest weighted error, at each step of the algorithm, the weak classifier selected to be added to the ensemble was the decision stump with the best split according to CART's Gini index splitting criteria (see Eq. 1.7 in §1.2.2). However, it was found that this would often not satisfy the weak learning condition. As the algorithm reached an iteration where such a weak classifier was chosen, reweighting would not occur and no more learning improvements would be made.

To solve this, the algorithm was updated to the exhaustive variant of AdaBoost presented in Fig. 1.8. At each step, the weak classifier with the minimum weighted error was chosen. This way, the weak learning condition was always satisfied and there were no breakdowns of the reweighting process.

2.4 Scikit-learn

The `sklearn` package for Python, known as scikit-learn, is an open-source machine learning toolbox built around the popular `scipy` package.[39] It contains optimised code for many popular machine learning algorithms, including implementations of decision tree classifiers and the AdaBoost algorithm used in this project, in the form of the classes `DecisionTreeClassifier` and `AdaBoostClassifier`.

The working of `AdaBoostClassifier` with `DecisionTreeClassifier` as the base learner is largely the same as the implementation used in this project, although with a few minor differences. `AdaBoostClassifier` supports a variation of the algorithm known as real AdaBoost that uses class probabilities in the reweighting process, alongside the discrete AdaBoost algorithm that we have implemented. `AdaBoostClassifier` also uses an estimation of the misclassified error rate when choosing the best weak classifier to be added to the ensemble with each iteration in order to be more computationally efficient, meaning it will not always choose the same weak classifiers.

Throughout this project, the performance of the AdaBoost implementation is compared to the performance of this package.

2.5 Evaluation Metrics

There are several ways in which the performance of a machine learning algorithm can be assessed. The simplest of these is looking at the training and validation error. This is simply the percentage of events in the training and validation dataset that are misclassified by the produced hypothesis. The lower the error, the better the performance of the algorithm. However, this often doesn't tell the whole story. A trained algorithm that looks to have a good performance based off these errors can have problems such as overfitting. The error also does not take into account how confident the predictions of the learning hypothesis are. If we had two situations where 60% of a dataset was correctly classified, but in one situations the hypothesis produced very confident predictions of positive or negative labels, while in the other situation the hypothesis produced predictions that were not very confident, we want to choose to favour the first hypothesis.

To improve on this, we look to make use of the confidence of AdaBoost's predictions, as discussed in §1.2.3. One of the ways to do this is to plot a histogram of these classifier predictions, for both the signal and background events, and for both the training and validation datasets. This allows us to see the distribution of how confident we can be in the predictions, and by comparing the histograms for the training and validation data, we can check for overfitting. If overfitting was occurring, we would expect to see a deviation between the two datasets, and if it is not we should see a similar distribution for both datasets.

We also utilise a plot known as a receiver operating characteristic (ROC curve). This is a plot of the true positive rate and false positive rate as the discrimination threshold is varied. The discrimination threshold is taken to be 0 when calculating classification errors, but here it is varied over the values generated by the classifier, with the true positive rate and false positive rate being calculated at each threshold. The true positive rate is the proportion of signal events correctly classified as such, while the false positive rate is the proportion of background events incorrectly classified as signal. The performance of a random guess is represented by a diagonal dotted line. Perfect classification would be achieved with a true positive rate of 1, and a false positive rate of 0. When assessing the performance of a classifier using an ROC curve, we look at the area under the curve (AUC), with a higher AUC equalling a better performance (AUC for a random guess would be 0.5, AUC for a perfect classifier would be 1). An example ROC curve can be seen in Fig. 2.6.

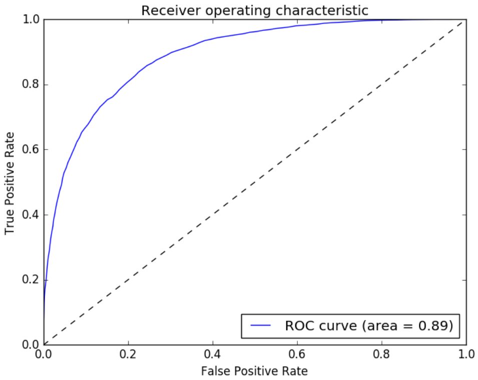


Figure 2.6: An example of a receiver operating characteristic plot.

3 Results & Discussion

In this section, we present and discuss the results of the investigations carried out using the AdaBoost algorithm. In §3.1 we look at the algorithm's performance on the $2\mu2e$ channel, and the $2\mu2\nu$ channel is covered in §3.2. For both channels, we look at the performance of the algorithm on the base parameter set and examine the effect of the number of splits used in creating the weak classifiers. Then we look at how the additional parameters affected the learning performance on each channel. In §3.3, we compare the results of our implementation to the results obtainable using the scikit-learn package.

3.1 $2\mu2e$ Channel

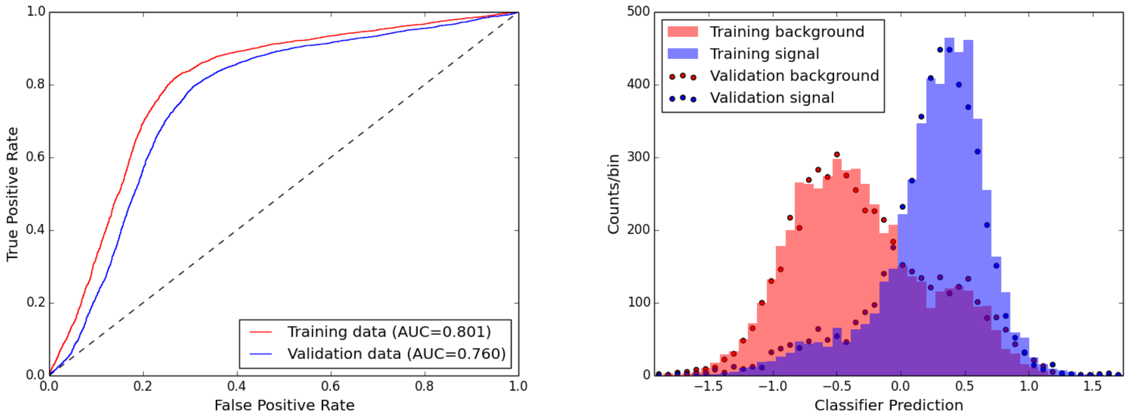
3.1.1 Base Parameter Set

The algorithm was run for 400 rounds of boosting, on the base parameter set, as described in §2.2.1. We first look at the difference in performance for differing numbers of splits used to construct the set of weak classifiers. The tested values were 10, 100, 1000, and 10,000 splits. The minimum errors for these different runs are shown in Table 1, along with the area under the curve for the training and validation ROC curves. We see that both the training error and validation error decrease with an increase in splits, and the AUC increases. The possibility of a large number of splits leading to overfitting to small trends in the training data seems to have been avoided. We conclude that performance on the $2\mu2e$ channel is best when the maximum number of splits, 10,000, is used to create the weak classifiers.

Number of Splits	10	100	1000	10000
Minimum Training Error	25.18%	23.09%	22.77%	22.51%
Training AUC	0.767	0.793	0.8	0.801
Minimum Validation Error	26.73%	25.57%	25.49%	25.28%
Validation AUC	0.746	0.758	0.758	0.76

Table 1: A table showing the results obtained using different numbers of splits on each parameter for the $2\mu2e$ channel.

The ROC curve and histogram plot are shown in Fig. 3.1 for the final hypothesis produced by the AdaBoost algorithm using 10,000 splits at the end of the 400 rounds of boosting. Looking at Fig. 3.1b, we see a similar distibution in the classifier prediction for both the training and validation datasets, further supporting the conclusion that overtraining is minimal as we reach this high number of splits. These results obtained using 10,000 splits, the best results obtained for the base parameter set, are the benchmark to which we will compare the performance of additional parameters.



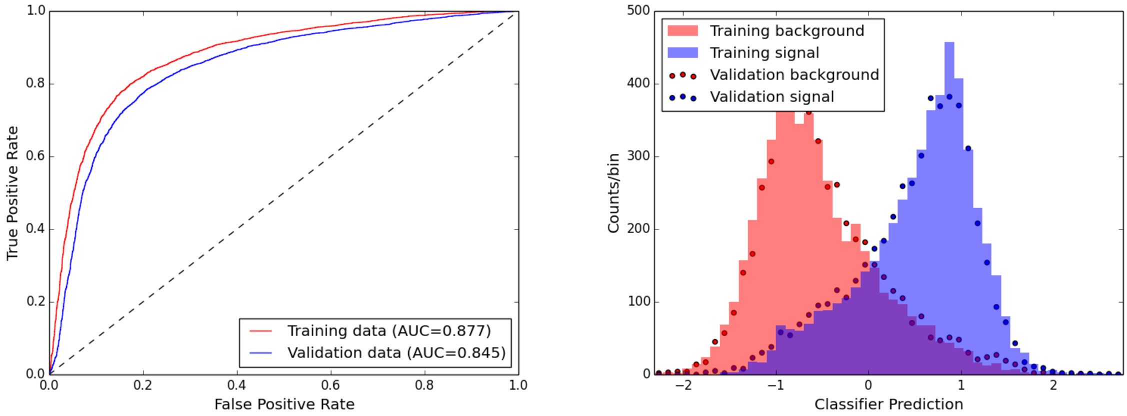
(a) The ROC curve

(b) Histogram plot of classifier output

Figure 3.1: The plots after 400 boosting iterations on the base parameter set on the $2\mu2e$ channel, with 10,000 splits on each parameter used to construct the weak classifiers.

3.1.2 Additional Parameters

Next, we look at the effect of adding the parameters `rap1`, `rap2`, `rap3`, `rap4`, `deltaRmu`, `deltaRe`, `m2mu`, and `m2e`, as discussed in §2.2.2. The results of 400 iterations of boosting with these additional parameters are displayed in Fig. 3.2. As with the base parameter set, we find similar trends in the errors and AUCs with an increase in splits, with 10,000 splits yielding the best performance.



(a) The ROC curve

(b) Histogram plot of classifier output

Figure 3.2: The plots for 400 boosting iterations, 10,000 splits, on the $2\mu 2e$ channel using the additional parameters.

A comparison of the performance of the additional parameters with the base parameter set is shown in Table 2, showing decreases in error, along with increases in AUC. Comparing the histogram plot (Fig. 3.2b) with the histogram plot for the base parameter set (Fig. 3.1b) shows an improvement in the classification of the background events, and an improved separation between signal and background. The improvement across all evaluation metrics leads us to conclude that the additional parameters introduced increase the predictive power of the algorithm.

10000 Splits	Additional Parameters	Base	Relative Improvement
Min. Training Error	18.77%	22.51%	-16.7%
Training AUC	0.877	0.801	+9.5%
Min. Validation Error	21.06%	25.28%	-16.7%
Validation AUC	0.845	0.760	+11.2%

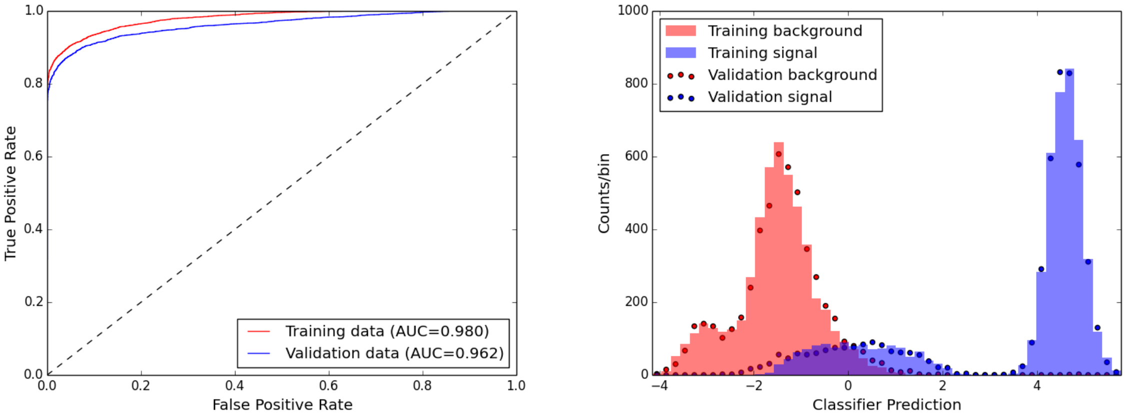
Table 2: A table showing the results for the additional parameters, and improvement from the base parameter set on the $2\mu 2e$ channel.

The results of training on the parameter `m4l`, in addition to the parameters discussed above, are summarised in Table 3 and Fig. 3.3. As with the previous cases, the run utilising 10000 splits is selected as the best result.

10000 Splits	m41	Additional Parameters	Relative Improvement
Min. Training Error	7.21%	18.77%	-61.5%
Training AUC	0.98	0.877	+11.7%
Min. Validation Error	8.30%	21.06%	-60.6%
Validation AUC	0.962	0.845	+13.8%

Table 3: A table showing the results for the additional parameter m41, and improvement from the previous parameter set (Table 2) on the $2\mu 2e$ channel.

The addition of this parameter further increases performance, to a best result of correctly classifying 91.7% of the validation sample. This corresponds to a 61.5% reduction in the error compared to the previous parameter set, and a total decrease in error of 67.1% when compared with the base parameter set. Examining the ROC curve in Fig. 3.3a, we see that both ROC curves rise to a true positive rate of 0.8 before the false positive rate starts to increase. This means that we correctly classify 80% of signal events before incorrectly labelling a background event. The histogram in Fig. 3.3b shows a clear separation, with the majority of signal given a very confident prediction by the classifier, with only a small fraction of the signal events having predictions overlapping with the predictions of the background events.



(a) The ROC curve

(b) Histogram plot of classifier output

Figure 3.3: The plots for 400 boosting iterations, 10,000 splits, on the $2\mu 2e$ channel using the additional parameter m41.

The high predictive power of this parameter is in line with our expectations given the strong peak observed in the distribution of the m41 parameter (Fig. 2.4). However, this good a performance is unlikely to be achievable on real data from a collider. Firstly, the pseudodata used to train the algorithm is simulated and contains precise values for the components of the 4-momenta for each particle, allowing the sharp peak in Fig. 2.4 to be constructed. The detectors of a collider cannot construct the 4-momentum with this same precision, and we would not expect to see such a defined signal when the reconstructed 4-momentum of all four leptons are combined. In addition to this matter of precision, our pseudodata consists of equal quantities of signal and background data. In reality, the background processes have a much greater cross section than the signal process. This would also make such a sharp signal harder to resolve. Further work would have to be carried out to see if the model we have trained on pseudodata would still produce accurate predictions on realistic collider data, and if not, if training on a realistic dataset would produce a model with similar predictive power.

3.2 $2\mu2\nu$ Channel

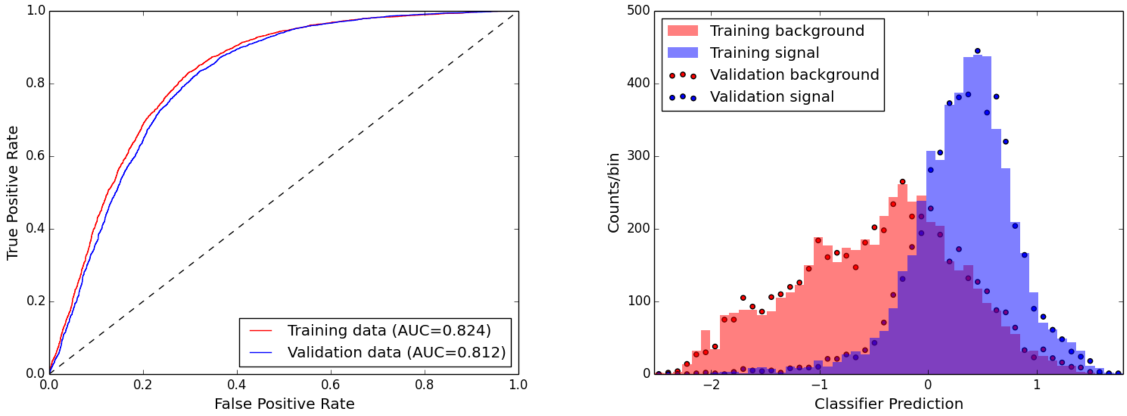
3.2.1 Base Parameter Set

The results for the base parameter set for the $2\mu2\nu$ channel, as described in §2.2.1, were compared for the different possible number of splits. As before, the values compared were 10, 100, 1000 and 10,000. The results are shown in Table 4As with the $2\mu2e$ channel, it was observed that the minimum training error decreased with an increased number of splits, while the training AUC increased. However, unlike the $2\mu2e$ channel, for the $2\mu2\nu$ channel the performance on the validation error and validation AUC improve as the number of splits increases from 10 to 100, but then this improvement reverses for a higher number of splits. As this number of splits is raised from 100 to 1000 and 10,000, the validation error begins to increase again, while the validation AUC decreases. This suggests that for this channel, a higher number of splits leads to overfitting, and the resulting hypotheses do not generalise well from the training dataset to the validation dataset. As such, we select the hypotheses generated using 100 splits as the best results. The ROC curve and histogram for these results are shown in Fig. 3.4a and Fig. 3.4b.

Number of Splits	10	100	1000	10000
Minimum Training Error	24.92%	23.61%	23.24%	22.99%
Training AUC	0.808	0.824	0.829	0.831
Minimum Validation Error	23.99%	23.92%	24.02%	24.29%
Validation AUC	0.809	0.812	0.810	0.809

Table 4: A table showing the results for algorithms trained using different numbers of splits on each parameter for the $2\mu2\nu$ channel.

Comparing the performance of these results to the performance on the base parameter set of $2\mu2e$, we see a lower minimum validation error of 23.92% for this channel, compared to 25.28% for the $2\mu2e$ channel. Comparing the ROC plot (Fig. 3.4a) to the ROC plot for the base parameter set (Fig. 3.1a), we also see that the AUCs are greater for the $2\mu2\nu$ channel. This seems counterintuitive at first, as we are able to construct all four final state particles for the $2\mu2e$ channel, whereas we could only construct two for the $2\mu2\nu$ channel due to the presence of the neutrinos. This effectively means the algorithm is fed half as much information for this channel. However, examining the distributions of the base parameters for $2\mu2e$ in Fig. 2.1 and $2\mu2\nu$ in Fig. 2.2, we saw that the parameters for $2\mu2\nu$ showed a better separation between signal and background than for $2\mu2e$. The fact that the signal and background events are better separated seems to lead to a stronger ensemble classifier, even with the reduced amount of data that is available to the algorithm.



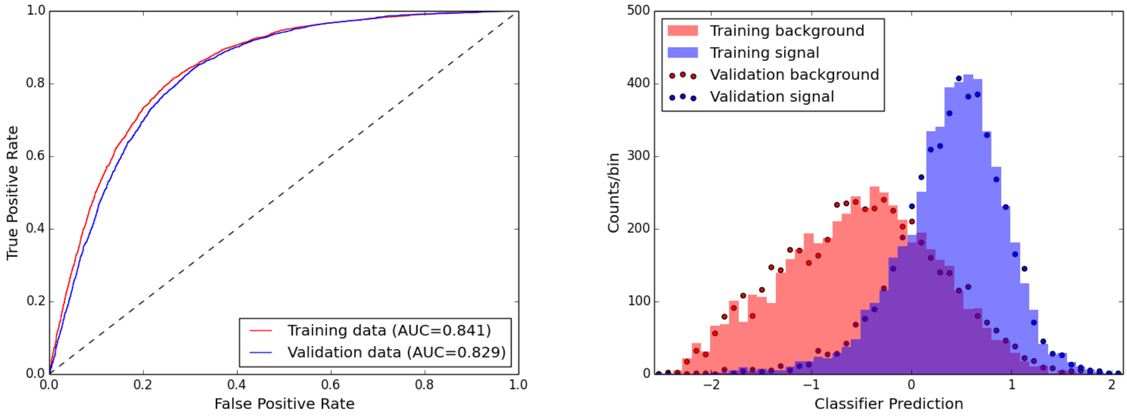
(a) The ROC curve

(b) Histogram plot of classifier output

Figure 3.4: The plots for 400 boosting iterations on the base parameter set on the $2\mu2\nu$ channel, with 100 splits on each parameter used to construct the weak classifiers.

3.2.2 Additional Parameters

The additional parameters examined here are `rap1`, `rap2`, and `deltaRmu`, as discussed in §2.2.2. The plots of these results are shown in Fig. 3.5, with the same overtraining effect being observed for a high number of splits, and 100 splits once again yielding the best performance when judged by both the error and AUC metrics.



(a) The ROC curve

(b) Histogram plot of classifier output

Figure 3.5: The plots for 400 boosting iterations on the $2\mu2\nu$ channel, with 100 splits on each parameter, using the additional parameters.

Table 5 shows a comparison of the performance with these additional parameters against the performance with the base parameter set. The addition of these parameters increases AUC performance when compared with the base parameter set, whilst decreasing both the training and validation error. However, the trend established in the base parameter set of a better performance on this channel than the $2\mu 2e$ channel does not carry over to the addition of these extra parameters, with the increases in performance being much smaller than those observed in §3.1.2 for the other channel. This is not unexpected, as we have only been able to construct 3 additional parameters as opposed to 8, and this did not include the reconstructed masses of the intermediate gauge bosons.

100 Splits	Additional Parameters	Base	Relative Improvement
Min. Training Error	22.56%	23.61%	-4.4%
Training AUC	0.841	0.824	+2.1%
Min. Validation Error	23.33%	23.92%	-2.5%
Validation AUC	0.824	0.812	+1.5%

Table 5: A table showing the results for the additional parameters, and improvement from the base parameter set on the $2\mu 2\nu$ channel.

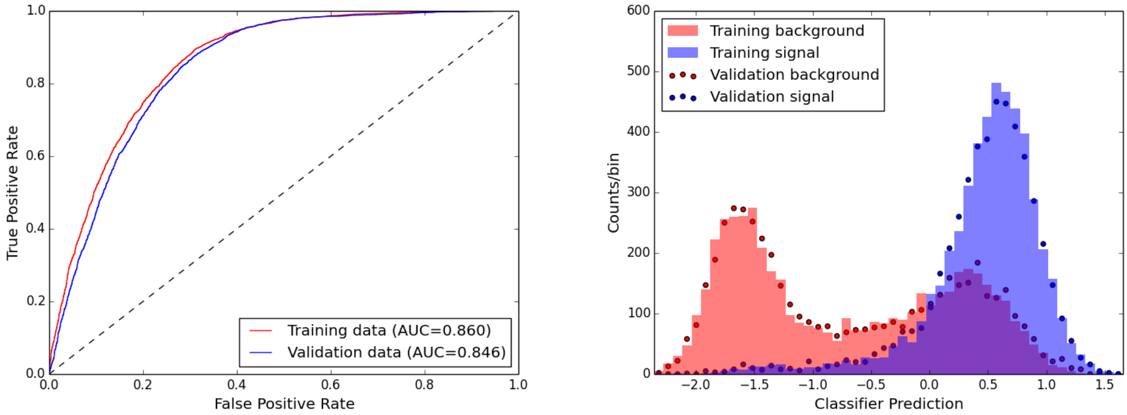
The introduction of `m2mu` proves to have a significant effect on the performance of the algorithm. A comparison of the results with and without this parameter are shown in Table 6. We see that with this additional parameter introduced, we achieve a best result of 21.65% validation error. This is a decrease in error of 7.2% from the previous parameter set and a total decrease of 9.5% from the base parameter set.

100 Splits	`m2mu`	Additional Parameters	Relative Improvement
Min. Training Error	20.89%	22.56%	-7.4%
Training AUC	0.860	0.841	+2.3%
Min. Validation Error	21.65%	23.33%	-7.2%
Validation AUC	0.846	0.824	+2.7%

Table 6: A table showing the results for the additional parameter `m2mu`, and improvement from the previous parameter set (Table 5) on the $2\mu 2\nu$ channel.

The substantial performance increase can also be seen from examining the histogram plot (Fig. 3.6b). We see that there is better separation between the signal and background predictions, with the introduction of a strong peak in background data not seen previously in the histogram for the base (Fig. 3.4b) or additional parameter set (Fig. 3.5b).

An increase in predictive power is introduced by this parameter for the $2\mu 2\nu$ channel, despite `m2mu` not having a direct physical significance in the same way `m2mu` and `m2e` reconstruct to the real/virtual Z boson in the $2\mu 2e$ channel. However, `m2mu` is subject to a constraint related to the Higgs mass. Even though we cannot fully construct the 4-particle invariant mass due to the momentum carried away by the neutrinos, the reconstructed mass of the muons from the signal process is strictly less than the invariant mass of the Higgs boson, as they have been produced from this initial state. This restriction does not apply to muons produced by background processes, and is enough of a discriminator that it increases the performance of our learning algorithm.



(a) The ROC curve

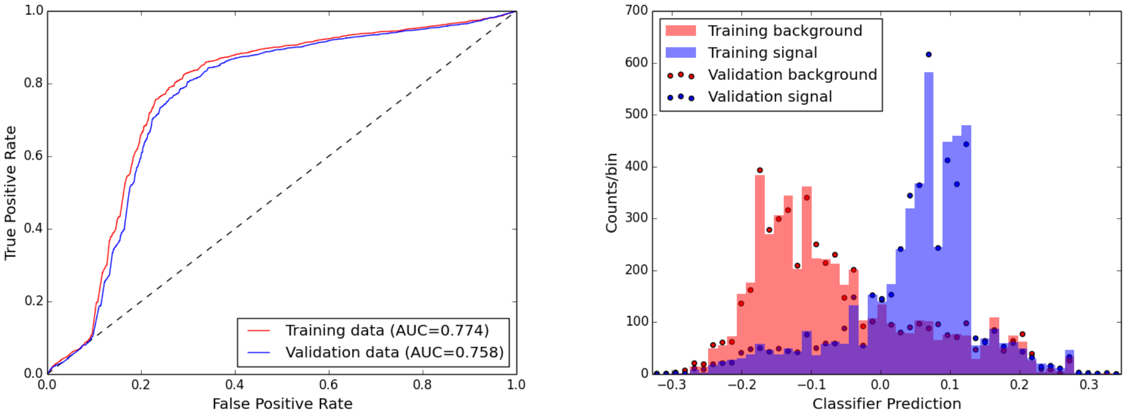
(b) Histogram plot of classifier output

Figure 3.6: The plots for 400 boosting iterations on the $2\mu2\nu$ channel, with 100 splits on each parameter, using the additional parameter `m2l`.

3.3 Comparison to Scikit-learn

We now compare the performance of our algorithm on the base parameter set for the $2\mu2e$ channel to the performance of `sklearn`. The ROC curve and histogram plot for `AdaBoostClassifier` are shown in Fig. 3.7. The `sklearn` implementation achieves minimum training and validation errors of 23.16% and 24.63%, compared to the values of 22.51% and 25.28% achieved by our implementation. Looking at the ROC curves (Fig. 3.1a and Fig. 3.7a), we see that our implementation achieves a greater training AUC than `sklearn`, along with a higher AUC for the validation data, although this is only a marginal difference (0.760 compared to 0.758), suggesting largely similar performance. Comparing the histograms (Fig. 3.1b and Fig. 3.7b), we see that a similar distribution in the classifier prediction is produced by both implementations for both the training and validation data, although the peaks are relatively higher for the `sklearn` implementation. Despite the similar performance when judged by the AUC metric, the validation error being lower by 0.6% leads us to conclude that `sklearn` outperforms our implementation for the base parameter set on this channel.

Examining the performance of `sklearn` on the additional rapidity, ΔR and reconstructed mass parameters, we see a similar situation to the base parameter set, with our implementation achieving marginally better AUC, but `sklearn` achieving a minimum validation error of 20.33%, 0.73% lower than our implementation. The same is true when looking at the `m4l` feature, with sklearn achieving a minimum validation error of 8.5%, compared with 8.61% for our implementation, and both implementations achieving an almost identical validation AUC. Thus, `sklearn`'s `AdaBoostClassifier` outperforms our implementation on both the additional and base parameter sets for the $2\mu2e$ channel.

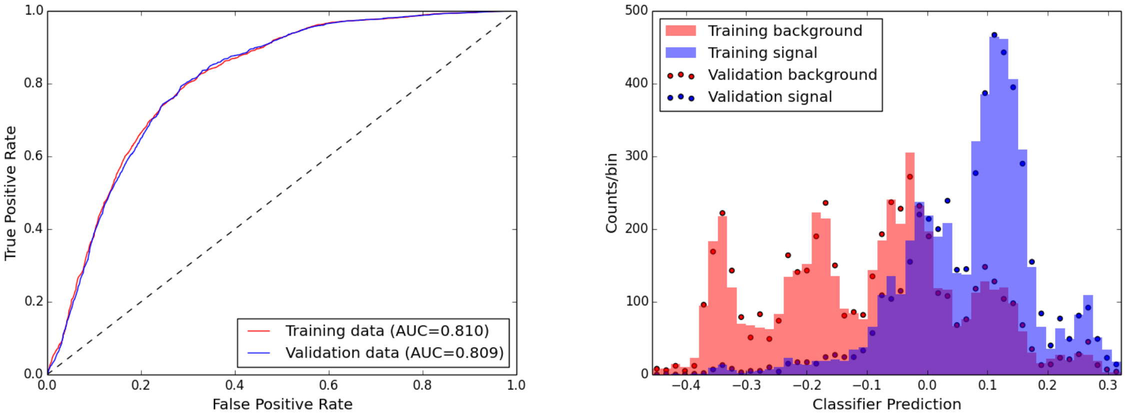


(a) The ROC curve

(b) Histogram plot of classifier output

Figure 3.7: The plots for 400 boosting iterations on the base parameter set of the $2\mu 2e$ channel, using the `sklearn` package.

Looking at the performance of `sklearn` on the $2\mu 2\nu$ channel, the ROC curve and histogram plots for `AdaBoostClassifier` can be seen in Fig. 3.8. This implementation does not perform as well as ours, achieving minimum training and validation errors of 24.98% and 24.56%, compared to values of 23.61% and 23.92% for our implementation. The AUC metric also yields better results for our implementation. Comparing the ROC curves (Fig. 3.4a and Fig. 3.8a), we see training and validation AUCs of 0.824 and 0.812 for our implementation, compared to values for `sklearn` of 0.810 and 0.809.



(a) The ROC curve

(b) Histogram plot of classifier output

Figure 3.8: The plots for 400 boosting iterations on the base parameter set of the $2\mu 2\nu$ channel, using the `sklearn` package.

This comparatively better performance for our implementation is also seen for the additional parameters, with our implementation yielding better results than `sklearn` for both the additional rapidity and ΔR parameters (validation error of 23.33%, validation AUC 0.829 vs. 23.74% and 0.825 for `sklearn`) and the `m2mu` parameter (validation error 21.65%, validation AUC 0.846 vs. 21.85% and 0.843 for `sklearn`).

4 Conclusions

In summary, in this project we have implemented an algorithm in order to distinguish between signal and background events on the $2\mu 2e$ and $2\mu 2\nu$ Higgs decay channels.

To do this, we have implemented a machine learning algorithm that generates decision stumps as weak learning rules, combining them using the AdaBoost algorithm. With each iteration, this selects the best weak classifier and adds it to the final classifier, reweighting the events in the training sample to concentrate more on incorrectly classified events.

We have trained the algorithm on 20,000 pseudodata events generated using MadGraph5_aMC@NLO to replicate final states at particle colliders. The performance of the algorithm was assessed using metrics of minimum training and validation errors, along with the area under the curve (AUC) of the receiver operating characteristic (ROC) plot.

We begun by investigating the performance of the algorithm on a base parameter set constructed from the 4-momentum of the final state particles. We concluded that the best number of splits to generate the decision stumps on each parameter on the $2\mu 2e$ channel is the maximum possible number, 10000. However, using this many splits leads to overfitting on the $2\mu 2\nu$ channel, and for this channel the best number of splits was found to be 100. The best number of splits was found to be the same on all additional parameter sets, as well as the base parameter sets, for both channels.

On the base parameter set, the algorithm achieved a minimum validation error and validation AUC of 25.28% and 0.76 for the $2\mu 2e$ channel. On the $2\mu 2\nu$ channel, we achieved results of 23.92% and 0.812, despite being unable to construct neutrino momenta.

We then added additional training parameters of rapidity and ΔR separation for both channels, along with the reconstructed masses of intermediate particles for the $2\mu 2e$ channel. This improved the results to an error of 21.06% and AUC of 0.845 for the $2\mu 2e$ channel, and 23.33% and 0.824 for the $2\mu 2\nu$ channel.

For the $2\mu 2e$ channel, we added the combined 4 lepton mass. This showed a sharp peak at 125GeV corresponding to the Higgs mass. This produced the best result, reducing the minimum validation error to 8.3% and increasing validation AUC to 0.962. However, we concluded the high predictive power may not carry over to real collider data, due to the unrealistic precision of the 4-momenta in the pseudodata, and because the pseudodata contains an equal split of signal and background, which wouldn't be the case for real data. Further work would be needed to see if similar results would be produced on real data.

For the $2\mu 2\nu$ channel, the combined invariant mass of the 2 muons reduced the error to 21.65%, and increased AUC to 0.846. This mass has no direct physical significance, but for signal events it is bounded above by the mass of the Higgs boson, so acts like a weaker version of the 4 lepton mass used to produce the best result on the $2\mu 2e$ channel.

We coompared our implementation of AdaBoost to the scikit-learn implementation, `AdaBoostClassifier`. For the $2\mu 2e$ channel, our implementation achieved marginally better validation AUC but scikit-learn achieved lower minimum validation error. On the $2\mu 2\nu$ channel, our implementation achieves both higher AUC and lower minimum error than the scikit-learn implementation on all tested combinations of training parameters.

References

[1] P. Higgs; *Broken Symmetries and the Masses of Gauge Bosons*; Physical Review Letters; Volume 13, Issue 16; pp. 508-509 (1964).

[2] P. Langacker; The Standard Model and Beyond; 1st Edition; Taylor and Francis; Boca Raton, FL, USA (2010).

[3] M. E. Peskin & D. V. Schroeder; An Introduction to Quantum Field Theory; 1st Edition; Perseus Books; New York, NY, USA (1995).

[4] T. Cheng & L. Li; Gauge Theory of Elementary Particle Physics; 1st Edition; Clarendon Press; Oxford, UK (1984).

[5] T. Muta; Foundation of Quantum Chromodynamics; 1st Edition; Singapore, Singapore (1984).

[6] S. L. Glashow; *The Renormalizability of Vector Meson Interactions*; Nuclear Physics; Volume 10; pp. 107-117 (1959).

[7] S. Weinberg; *A Model of Leptons*; Physical Review Letters; Volume 19, Issue 21; pp. 1264-1266 (1967).

[8] A. Salam; *Weak and Electromagnetic Interactions*; Il Nuovo Cimento; Volume 11, Issue 4; pp.568-577 (1959).

[9] C. Quigg; *Spontaneous Symmetry Breaking as a Basis of Particle Mass*; Reports on Progress in Physics; Volume 70; pp. 1019-1054 (2007).

[10] A. Djouadi; *The Anatomy of Electro-Weak Symmetry Breaking: The Higgs Boson in the Standard Model*; Physics Reports; Volume 457; pp. 1-216 (2008).

[11] *The LHC's new frontier*; CERN Courier; 5 May (2010).

[12] G. Aad et al. (ATLAS Collaboration); *Observation of a New Particle in the Search for the Standard Model Higgs Boson with the ATLAS Detector at the LHC*; Physics Letters B.; Volume 716, Issue 1; pp. 1-29 (2012).

[13] S. Chatrchyan et al. (CMS Collaboration); *Observation of a New Boson at a Mass 125 GeV with the CMS Experiment at the LHC*; Physics Letters B.; Volume 716, Issue 1; pp. 30-61 (2012).

[14] *Stable beams at 13 TeV*; CERN Courier; 22 July (2015).

[15] D. A. van Dyk; *The Role of Statistics in the Discovery of a Higgs Boson*; The Annual Review of Statistics and Its Application 2014; Volume 1; pp. 41-59 (2014).

[16] G. Cowan; Statistical Data Analysis; 1st Edition; Clarendon Press; Oxford, UK (1998).

[17] G. Aad et al. (ATLAS Collaboration); *The ATLAS Experiment at the CERN Large Hadron Collider*; Journal of Instrumentation; JINST 3 S08003 (2008).

[18] S. Chatrchyan et al. (CMS Collaboration); *The CMS Experiment at the CERN LHC*; Journal of Instrumentation; JINST 3 S08004 (2008).

[19] E. Torrence; *Luminosity Public Results Run 2*; twiki.cern.ch; Available at: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2 [Accessed 15th Jan 2018].

[20] T. Han; *Collider Phenomonology: Basic Knowledge and Techniques*; Lectures Given at Conference; Conference Proceedings C04-06-06.1; pp. 407-454 (2005).

[21] J. Baglio, A. Djouadi & J. Quevillon; *Prospects for Higgs physics at energies up to 100 TeV*; Reports on Progress in Physics; Volume 79, Issue 11 (2016).

[22] G. Aad et al. (ATLAS Collaboration); *Measurements of four-lepton production in pp collisions at* $\sqrt{s} = 8$ *TeV with the ATLAS detector*; Physics Letters B.; Volume 753, Issue 1; pp. 552-572 (2016).

[23] P. Achard et al. (L3 Collaboration); *Measurement of the cross section of W-boson pair production at LEP*; Physics Letters B.; Volume 600, Issue 1; pp. 22-40 (2004).

[24] T. Binoth et al.; *Gluon-induced W-boson pair production at the LHC*; Journal of High Energy Physics; JHEP 0612:046 (2006).

[25] T. M. Mitchell; Machine Learning; 1st Edition; McGraw-Hill; New York, NY, USA; p. 2 (1997).

[26] T. M. Mitchell; Machine Learning; 1st Edition; McGraw-Hill; New York, NY, USA; p. 53 (1997).

[27] J. R. Quinlan; *Induction of Decision Trees*; Machine Learning; Volume 1; pp. 81-106 (1986).

[28] L. Breiman et al.; Classification and Regression Trees; 1st Edition; Chapman & Hall; Boca Raton, FL, USA (1984).

[29] S. Raschka & V. Mirjalili; Python Machine Learning; 2nd Edition; Packt; Birmingham, UK; pp. 90-94 (2017).

[30] C. Zhang & Y. Ma; Ensemble Machine Learning; 1st Edition; Springer; New York, NY, USA (2012).

[31] Y. Freund & R. E. Schapire; *A decision-theoretic generalization of on-line learning and an application to boosting*; Journal of Computer and System Sciences; Volume 55, Issue 1; pp. 119-139 (1997).

[32] Y. Freund & R. E. Schapire; *Experiments with a New Boosting Algorithm*; Proceedings of the Thirteenth International Conference on Machine Learning; pp. 148-156 (1996).

[33] Y. Freund & R. E. Schapire; *Improved Boosting Algorithms Using Confidence-rated Predictions*; Machine Learning; Volume 37, Issue 3; pp. 297-336 (1999).

[34] R. E. Schapire; *Explaining Adaboost*; Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik; 1st Edition; Springer; Berlin, Germany; pp. 37-52 (2013).

[35] R. E. Schapire & Y. Freund; Boosting: Foundations and Algorithms; 1st Edition; MIT Press; Cambridge, MA, USA (2012).

[36] J. Alwall et al.; *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*; Journal of High Energy Physics; JHEP 1407:079 (2014).

[37] J. Alwall et al.; *A standard format for Les Houches Event Files*; Computer Physics Communications; Volume 176, Issue 4; pp. 300-304 (2007).

[38] J. Beringer et al. (Particle Data Group); *Review of Particle Physics*; Physical Review D.; Volume 86, Issue 1; pp. 415-418 (2012).

[39] F. Pedragosa et al.; *Scikit-learn: Machine Learning in Python*; Journal of Machine Learning Research; Volume 12, Issue 10 (Oct); pp.2825-2830 (2011).

I Appendix - Theoretical Derivation of the Higgs Mechanism

Symmetries of the Standard Model

The internal symmetries of the standard model are $SU(3)_c$, $SU(2)_L$, and $U(1)_Y$. Each of these symmetries can be represented as a Lie group.[1] Each group has an associated Lie algebra, the elements of which are infinitesimal generators of the group.

$U(1)$ is the unitary group, containing all complex numbers of modulus 1 under multiplication, so the physical representation of this group is phase rotations on a unit circle.

$SU(2)$ and $SU(3)$ are special unitary groups, corresponding to the Lie groups of 2x2 and 3x3 unitary matrices with unit determinant under matrix multiplication.

$SU(2)$ is generated by $i\sigma_i$, where σ_i are the Pauli matrices:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Similarly, $SU(3)$ is generated by λ_i, where λ_i are the Gell-Mann matrices (a generalisation of the Pauli matrices):

$$\lambda_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \lambda_2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ o & 0 & 0 \end{pmatrix} \quad \lambda_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \lambda_4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$\lambda_5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix} \quad \lambda_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \lambda_7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix} \quad \lambda_8 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}$$

We now return to the standard model, and the full $SU(3)_c \times SU(2)_L \times U(1)_Y$ symmetry. Here c denotes *colour*, L denotes *left*, and Y denotes *hypercharge*. Each of these symmetries has a physical importance. According to Noether's Theorem, each of these gauge symmetries leads to a conserved quantity.[2] Colour charge is the conserved quantity for $SU(3)_c$ symmetry, while the conserved quantity for $SU(2)_L$ symmetry is weak hypercharge and $U(1)_Y$ symmetry is associated with the conservation of electric charge.

Each of the gauge groups have a corresponding gauge field, and an associated force.

$SU(3)_c$ is the gauge group associated with the strong force, and the force carrier is the gluon. The 8 Gell-Mann matrices which are the generators of the $SU(3)_c$ group are equivalent to the 8 independent colour states of the gluons.

The $SU(2) \times U(1)_Y$ group corresponds to a unified electroweak gauge field. This unified group has 4 generators. However, unlike $SU(3)_c$, this symmetry is not preserved; it is said to be spontaneously broken. The spontaneous breaking of this $SU(2)_L \times U(1)_Y$ symmetry to $U(1)_{el}$ symmetry is due to the Higgs mechanism.[9]

Spontaneous Symmetry Breaking & The Higgs Mechanism

A field theory is said to have a spontaneously broken symmetry when the Lagrangian $\mathcal{L}$ is invariant under a given symmetry transformation, but the vacuum state (ground state) is not invariant with respect to the same symmetry. We first examine the general case of spontaneous symmetry breaking for a continuous symmetry, before considering the specific case of the standard model.[3]

Consider a complex-valued scalar field $\phi = \phi_1 + i\phi_2$ with the following Lagrangian:

$$\mathcal{L} = (\partial_\mu \phi)^*(\partial^\mu \phi) - \mu^2 \phi^* \phi - \lambda(\phi^* \phi)^2, \lambda > 0. \tag{I.1}$$

This Lagrangian has $U(1)$ global symmetry. That means it is invariant under the transformation:

$$\phi(x) \to \phi'(x) = e^{-i\theta}\phi(x). \tag{I.2}$$

We now seek to minimise the potential in order to obtain the vacuum expectation value (VEV):

$$\frac{\partial V}{\partial \phi} = \mu^2 \phi^* + 2\lambda(\phi^* \phi)^2 \phi^* = 0, \tag{I.3}$$

giving us the following condition:

$$\phi^* \phi = |\phi|^2 = \frac{-\mu^2}{2\lambda} \equiv \frac{v^2}{2}. \tag{I.4}$$

For $\mu^2 > 0$, the potential has a minimum at $\phi = 0$. Hence the VEV, $\langle \phi \rangle_0 = 0$. In this case the global $U(1)$ symmetry is preserved and there is no spontaneous symmetry breaking.

For $\mu^2 < 0$, we obtain a continuum of degenerate minima of the potential, with VEV satisfying:

$$|\langle \phi \rangle_0| = \sqrt{\frac{-\mu^2}{2\lambda}} \equiv \frac{v}{\sqrt{2}}. \tag{I.5}$$

Therefore the VEV is not invariant under $U(1)$ global symmetry transformations. So the $U(1)$ symmetry is spontaneously broken. The continuum of minima corresponds to a circle, and the visualisation of the situation is shown in Fig. I.1.

We now choose the VEV to be along the real axis, so $\phi = \phi_1$:

$$\langle \phi \rangle_0 = \langle \phi_1 \rangle_0 = \frac{v}{\sqrt{2}}. \tag{I.6}$$

Then we expand the Lagrangian around the vacuum state, expressing these fluctuations as $\phi = \frac{1}{\sqrt{2}}(v + \xi + i\chi)$. The expanded Lagrangian becomes:

$$\mathcal{L} = \frac{1}{2}\partial_\mu \xi \partial^\mu \xi + \frac{1}{2}\partial_\mu \chi \partial^\mu \chi - \lambda v^2 \xi^2 - \lambda v \xi(\xi^2 + \chi^2) - \frac{1}{4}(\xi^2 + \chi^2)^2 + const. \tag{I.7}$$

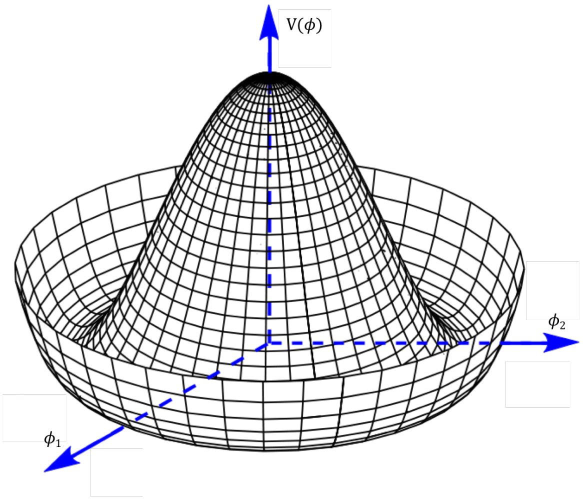


Figure I.1: The 'mexican hat' potential

Examining the quadratic term in ξ, $-\lambda v^2\xi^2$, and comparing it to a general potential term $V = \frac{1}{2}m^2x^2$, we can conclude that ξ is a massive real scalar field, corresponding to radial fluctuations around the vacuum state, with mass:

$$m_\xi = \sqrt{-2\mu^2} \equiv \sqrt{2\lambda}v. \tag{I.8}$$

As there is no quadratic term in χ, we conclude that this is a massless field corresponding to oscillations around the vacuum state in the tangential direction.

The appearance of massless fields/particles is a general property of spontaneous symmetry breaking. In fact, every broken symmetry corresponds to a massless particle. This is formally stated by Goldstone's Theorem[2]:

If a continuous global symmetry is spontaneously broken, the theory must contain a massless particle for each generator of the broken symmetry group.

The massless particles that arise as a result of such spontaneous symmetry breaking are called *Goldstone bosons*. These massless particles exist only when considering global symmetries, not gauge symmetries. When one considers a gauge symmetry, the Goldstone bosons disappear, and the gauge bosons associated with each generator of the broken symmetry become massive. Gauge bosons associated with unbroken symmetries remain massless. The gauge bosons can be considered to have 'absorbed' the Goldstone bosons, along with their associated degrees of freedom. This phenomenon of bosons acquiring mass from spontaneously broken symmetries is known as the *Higgs mechanism*.

The Standard Model Higgs Mechanism

The model that provides the description of the electroweak interaction that fits correctly with experimental data was introduced by Glashow, Weinberg and Salam.[4][5][6] It is a non-Abelian gauge theory with $SU(2)$ and $U(1)$ gauge symmetry. These are the symmetries previously discussed in §I.

The model consists of an $SU(2)$ gauge field coupled to a complex-valued scalar field $\phi = \phi_1 + i\phi_2$ that transforms as a spinor of $SU(2)$, and imposing the $U(1)$ gauge symmetry gives an overall gauge transformation of:

$$\phi(x) \to \phi'(x) = e^{i\alpha^a \tau^a} e^{i\frac{\beta}{2}} \phi(x). \tag{I.9}$$

Here, $\tau^a = \frac{\sigma^a}{2}$, where σ^a are the Pauli matrices, as described in §I.

The full Lagrangian for the standard model has many terms, so we look only at the terms that are relevant for spontaneous symmetry breaking:

$$\mathcal{L}_\phi = (D_\mu \phi)^\dagger (D^\mu \phi) - V(\phi^\dagger \phi) \tag{I.10}$$

The covariant derivative of ϕ is:

$$D_\mu \phi = (\partial_\mu - igA^a_\mu \tau^a - \frac{i}{2} g' B_\mu)\phi, \tag{I.11}$$

where A^a_μ are the 3 $SU(2)$ gauge bosons, B_μ is the $U(1)$ gauge boson, and g and g' are their respective coupling constants. The potential term is:

$$V(\phi^\dagger \phi) = \mu^2 \phi^\dagger \phi + \lambda(\phi^\dagger \phi)^2. \tag{I.12}$$

We choose a real value for the vacuum expectation value to be:

$$\langle \phi \rangle_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v \end{pmatrix}, \tag{I.13}$$

which is not invariant under $SU(2)$ gauge transformations. We consider the action of the following linear combination of generators on the VEV:

$$Q\langle \phi \rangle_0 = (T^3 + Y)\langle \phi \rangle_0 = 0 \quad K\langle \phi \rangle_0 = (T^3 - Y)\langle \phi \rangle_0 \neq 0 \quad T^1 \langle \phi \rangle_0 \neq 0 \quad T^2 \langle \phi \rangle_0 \neq 0, \tag{I.14}$$

and see that three gauge bosons have spontaneously acquired mass. The 1st operator, Q, corresponding to the electric charge, annihilates the vacuum, so we see that the $U(1)$ symmetry of electromagnetism is preserved. The boson associated with this symmetry, the photon, remains massless. The other 3 operators lead to a non-zero vacuum expectation value. We say that the $SU(2)_L \times U(1)_Y$ symmetry has broken down to $U(1)_{el}$ symmetry.

As in §I, we expand the Lagrangian around the vacuum state, in the unitary gauge, this time expressing these fluctuations as:

$$\phi = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v + \xi \end{pmatrix} \tag{I.15}$$

.

Then the Lagrangian becomes:

$$\mathcal{L}_\phi = \frac{1}{2}(\partial_\mu \xi)^2 + \frac{v^2}{8}[g^2 (A^1_\mu)^2 + g^2 (A^2_\mu)^2 + (gA^3_\mu - g' B_\mu)^2] - \mu^2 \xi^2 - \lambda v \xi^3 - \lambda 4 \xi^4 \tag{I.16}$$

Examining the quadratic term in ξ, similarly to §I, we can conclude that ξ is a massive real scalar field with mass:

$$m_\xi = \sqrt{-2\mu^2} \equiv \sqrt{2\lambda}v. \tag{I.17}$$

This scalar field is called the Higgs field,[7] and we see that it couples to itself, providing the Higgs boson with a mass.

In order to also find the mass of the gauge bosons, we seek to rewrite the Lagrangian in terms of the mass eigenstate fields $W_\mu^\pm$, Z_μ^0, A_μ, rather than the gauge fields A_μ^a, B_μ. To do this we write $W_\mu^\pm = \frac{1}{\sqrt{2}}(A_\mu^1 \mp iA_\mu^2)$ and perform a change of basis:

$$\begin{pmatrix} Z_\mu^0 \\ A_\mu \end{pmatrix} = \begin{pmatrix} \cos\theta_w & -\sin\theta_w \\ \sin\theta_w & \cos\theta_w \end{pmatrix} \begin{pmatrix} W_\mu^3 \\ B_\mu \end{pmatrix}, \tag{I.18}$$

where θ_w is called the weak mixing angle, with $\tan\theta_w = \frac{g}{g'}$.

The $(D_\mu\phi)^\dagger(D^\mu\phi)$ term of the Lagrangian then becomes:

$$(D_\mu\phi)^\dagger(D^\mu\phi) = \frac{1}{2}(\partial_\mu\xi)^2 + \frac{m_W^2}{2}(|W_\mu^+|^2 + |W_\mu^-|^2) + \frac{m_Z^2}{2}|Z_\mu^0|^2 + 0A_\mu A^\mu \tag{I.19}$$

Comparing Eq. I.19 with Eq. I.16, we conclude that:

$$m_{W^\pm} = \frac{gv}{2}, \quad m_Z = \frac{1}{2}v\sqrt{g^2 + g'^2}, \quad m_A = 0. \tag{I.20}$$

We have derived the masses of the two W bosons, the Z boson and the photon. We have a massless photon, and expressions for the W, Z and Higgs bosons in terms of the Higgs vacuum expectation value, v. Experimentally, this is found to be 246GeV,[8] giving approximate values of the masses:

$$m_{W^\pm} = 80\text{GeV}, \quad m_Z = 91\text{GeV}, \quad m_\xi = 125\text{GeV}, \tag{I.21}$$

where we use natural units and the mass-energy relation $E = \sqrt{p^2c^2 - m^2c^4}$ in order to express masses in terms of energy, in units of electron volts ($1eV = 1.60 \times 10^{-19}$J).

Higgs-Fermion Coupling

In order to derive the above gauge boson masses, we examined the $\mathcal{L}_\phi$ term of the standard model Lagrangian. The full Lagrangian also contains $\mathcal{L}_{Y,L}$ and $\mathcal{L}_{Y,Q}$ terms, known as Yukawa couplings, which result in the fermions also acquiring masses from interaction with the vacuum Higgs field. All fermions couple to the Higgs field, with a mass proportional to the Higgs VEV, v.

We examine as an example the Yukawa coupling term for the electron:

$$\mathcal{L}_{Y,e} = -\lambda_e(\bar{L}\cdot\phi R + \bar{R}\phi^\dagger\cdot L). \tag{I.22}$$

Expanding ϕ around the VEV we get a spontaneously broken symmetry giving mass to the electron:

$$\mathcal{L}_{Y,e} \simeq -\frac{\lambda_e v}{\sqrt{2}}(\bar{e_L}e_R + \bar{e_R}e_L) = -m_e\bar{e}e, \quad m_e = \frac{\lambda_e v}{\sqrt{2}} \tag{I.23}$$

The coupling is the same for the other generations of leptons, with different factors λ_L.

For the quark sector, the Yukawa terms mix quarks of different generations, forming complex valued matrices. The physical particles are those that diagonalise the matrix. The mass eigenstates can be related to the gauge eigenstates through the Cabbibo-Kobayashi-Maskawa (CKM) matrix.

References

[1] A. Kirillov; Introduction to Lie Groups and Lie Algebras; 1st Edition; Cambridge University Press; Cambridge, UK (2008).

[2] M. E. Peskin & D. V. Schroeder; An Introduction to Quantum Field Theory; 1st Edition; Perseus Books; New York, NY, USA (1995).

[3] C. Zambon; *Gauge Field Theories*; Lecture Notes; Particle Theory PHYS4181; University of Durham; delivered January 2018.

[4] S. L. Glashow; *The Renormalizability of Vector Meson Interactions*; Nuclear Physics; Volume 10; pp. 107-117 (1959).

[5] S. Weinberg; *A Model of Leptons*; Physical Review Letters; Volume 19, Number 21; pp. 1264-1266 (1967).

[6] A. Salam; *Weak and Electromagnetic Interactions*; Il Nuovo Cimento; Volume 11, Number 4; pp.568-577 (1959).

[7] C. Quigg; *Spontaneous Symmetry Breaking as a Basis of Particle Mass*; Reports on Progress in Physics; Volume 70; pp. 1019-1054 (2007).

[8] J. Beringer et al. (Particle Data Group); *Review of Particle Physics*; Physical Review D.; Volume 86; Issue 1 (2012).