Boosted Regression Tree Based Invariant Mass Reconstruction of τ Lepton Pairs at ATLAS

October 1, 2014

Matthew Quenneville
Supervisor: Dr. Dugan O'Neil

Abstract

Boosted regression trees were tested as an alternative to the currently used invariant mass reconstruction algorithms for hadronically decaying tau lepton pairs. In particular, focus was placed on the mass range necessary for accurate measurement of the Higgs Boson. Over this range, boosted regression trees showed performance comparable to that of the currently used algorithms, but using only a small fraction of computational resources for evaluation.

Contents

1 Introduction

In July 2012, the ATLAS and CMS collaborations announced the discovery of a new particle, which has since been identified as the Higgs boson [1] [2]. Measurements of this particle have indicated that it agrees very well with the predictions of the Standard Model. However, in order to test the validity of the theory, it is important to measure the Higgs boson's various decay channels. Of particular interest is its coupling to the tau leptons. As the heaviest leptons, taus give the best opportunity to observe one of the Higgs' predicted leptonic couplings. The decay channel of interest for this measurement is the channel in which the Higgs decays into a tau and its corresponding anti-particle. Taus are not as simple to detect as the other charged leptons. They decay almost instantly, and all of their decays involve at least one neutrino, which is not measured. Thus, it is not possible to fully reconstruct the momentum vectors of taus. This implies that the invariant mass of a tau anti-tau pair cannot be directly computed. This is further complicated by the fact that the Z boson can also decay to a tau anti-tau pair, resulting in a background which is much larger than the Higgs signal, and only differentiable due to the invariant mass. Thus, it is essential for such an analysis to have a method of estimating the invariant mass using only the observable quantities. Due to the enormous amount of data involved in such an analysis, ideally, this technique would also be quick to evaluate. This paper presents an alternative method of approximating the mass of tau lepton pairs which is competitive with the best currently used algorithm in terms of performance, but orders of magnitude faster to compute.

2 ATLAS

ATLAS is a detector on the main ring of the Large Hadron Collider (LHC) in Geneva, Switzerland. The LHC collides bunches of protons at center of mass energies of up to 8 TeV. ATLAS detects the particles produced in such collisions. ATLAS has a cylindrical geometry, surrounding the beam line. The beam line is taken to define the z axis of the coordinate system. The point along the z axis exactly in the geometric center of the detector is taken to define $z = 0$. Due to the azimuthal symmetry of the detector, the x and y axes can be taken to be anywhere in the plane perpendicular to z (the transverse plane), so long as they are orthogonal, and form a right handed coordinate system. The azimuthal angle, ϕ, is then defined to be with respect to the x axis, such that the y axis lies along $\phi = \pi/2$. Due to the reflection symmetry of the detector about $z = 0$, one direction can be chosen arbitrarily to define the positive z axis. A polar angle, θ, relative to the positive z axis is then defined. Since differences in θ are not lorentz invariant, a new angular coordinate is usually used in its place, called the pseudorapidity. The pseudorapidity, η, is defined by $\eta = -\ln(\tan(\frac{\theta}{2}))$.

The ATLAS detector is able to detect particles that interact via the electromagnetic and strong interactions. Therefore, neutrinos pass through the detector undetected. However, some information about their trajectories can be recovered. Since momentum must conserved in the transverse plane, and the initial protons have no momentum in this plane, the vector sum of the transverse momenta of all final state particles must add to 0. However, if neutrinos are released, the momenta measured in this plane will be unbalanced. Therefore, the transverse component of the vector sum of all invisible objects (neutrinos in this case), can be estimated by measuring the amount of missing momentum that is needed to balance a given event. This is referred to as the missing transverse momentum, or $\not{E}_T$.

3 Currently Used Algorithms

Perfect reconstruction of the invariant mass of a ditau resonance would require solving for 8 unknowns. These 8 are the missing four-momentum components of the two tau decays. In other words, the complete

tau four-vectors, $p^{\mu}_{\tau_1}$ and $p^{\mu}_{\tau_2}$, are given by:

$$p^{\mu}_{\tau_i} = p^{\mu}_{\tau_i,\text{vis}} + p^{\mu}_{\tau_i,\text{mis}} \tag{1}$$

where the subscript vis refers to the visible part of the tau decay, and the subscript mis refers to the missing, or invisible, part. Thus, $p^{\mu}_{\tau_i,\text{vis}}$ are taken to be measureable, and $p^{\mu}_{\tau_i,\text{mis}}$ cannot be directly measured. By conservation of momentum, since there is no momentum transverse to the beam line before the collision, we expect all transverse momenta to sum to 0 in an event. The missing transverse energy, $\not{E}_T$, can be defined as negative the vector sum of the transverse momenta in the event. A large $\not{E}_T$ suggests that some particle may have passed through the detector undetected. If we assume $\not{E}_T$ is entirely due to the neutrinos, this can be taken to be the vector sum of the transverse momenta of the neutrinos in the event. If a tau decays hadronically, there is only one neutrino released, and thus, m_{mis}, the mass component of the invisible tau four-vector, is given by $m_{\text{mis}} = m_{\nu_\tau} \approx 0$, reducing the number of unknowns for that tau from 4 to 3. Thus, for $H \to \tau_{\text{lep}}\tau_{\text{lep}}$ there are 8 unknowns, for $H \to \tau_{\text{lep}}\tau_{\text{had}}$ there are 7 unknowns, and for $H \to \tau_{\text{had}}\tau_{\text{had}}$ there are 6.

There are four equations connecting these unknowns from kinematic constraints:

$$\begin{aligned}
\not{E}_{T,x} &= p_{1,\text{mis}} \sin\theta_{1,\text{mis}} \cos\phi_{1,\text{mis}} + p_{2,\text{mis}} \sin\theta_{2,\text{mis}} \cos\phi_{2,\text{mis}} \\
\not{E}_{T,y} &= p_{1,\text{mis}} \sin\theta_{1,\text{mis}} \sin\phi_{1,\text{mis}} + p_{2,\text{mis}} \sin\theta_{2,\text{mis}} \sin\phi_{2,\text{mis}} \\
M^2_{\tau_1} &= m^2_{1,\text{mis}} + m^2_{1,\text{vis}} + 2\sqrt{p^2_{1,\text{vis}} + m^2_{1,\text{vis}}}\sqrt{p^2_{1,\text{mis}} + m^2_{1,\text{mis}}} - 2p_{1,\text{vis}}p_{1,\text{mis}}\cos(\theta_{1,\text{mis}} - \theta_{1,\text{vis}}) \\
M^2_{\tau_2} &= m^2_{2,\text{mis}} + m^2_{2,\text{vis}} + 2\sqrt{p^2_{2,\text{vis}} + m^2_{2,\text{vis}}}\sqrt{p^2_{2,\text{mis}} + m^2_{2,\text{mis}}} - 2p_{2,\text{vis}}p_{2,\text{mis}}\cos(\theta_{2,\text{mis}} - \theta_{2,\text{vis}})
\end{aligned} \tag{2}$$

where $\not{E}_{T,x}$ and $\not{E}_{T,y}$ are the x and y components of $\not{E}_T$. The momentum of the invisible component of τ_i is $p_{i,\text{mis}}$, with direction defined by polar angle $\theta_{i,\text{mis}}$ and azimuthal angle $\phi_{i,\text{mis}}$. The visible tau four-vector is similarly defined by the momentum, mass, polar angle, and azimuthal angle $p_{i,\text{vis}}$, $m_{i,\text{vis}}$, $\theta_{i,\text{vis}}$, and $\phi_{i,\text{vis}}$ respectively. M_{τ_i} are taken to be the tau lepton mass ($M_{\tau_i} \approx 1.777$ GeV). With four equations, but 6-8 unknowns, these equations are under-constrained. Thus, they cannot be directly solved for the mass. There are two variables of interest which use different methods to get around this issue. These two that will be used for comparison are the collinear mass, and the Missing Mass Calculator (MMC) mass.

3.1 Collinear Mass

The collinear mass is a commonly used variable for $H \to \tau\tau$ analysis, thanks to its good performance under certain conditions, and its relative simplicity to compute. It is based around the assumption that the neutrinos resulting from the tau decays are travelling in approximately the same directions as the corresponding visible tau decay products. Mathematically this says:

$$\begin{aligned}
\theta_{1,\text{mis}} &= \theta_{1,\text{vis}} \\
\phi_{1,\text{mis}} &= \phi_{1,\text{vis}} \\
\theta_{2,\text{mis}} &= \theta_{2,\text{vis}} \\
\phi_{2,\text{mis}} &= \phi_{2,\text{vis}}
\end{aligned} \tag{3}$$

Thus, with 8 equations and at most 8 unknowns, the equations can be solved analytically. The system of equations (2) simply reduces to solving the following for $p_{1,\text{mis}}$ and $p_{2,\text{mis}}$:

$$\begin{aligned}
\not{E}_{T,x} &= p_{1,\text{mis}} \sin\theta_{1,\text{vis}} \cos\phi_{1,\text{vis}} + p_{2,\text{mis}} \sin\theta_{2,\text{vis}} \cos\phi_{2,\text{vis}} \\
\not{E}_{T,y} &= p_{1,\text{mis}} \sin\theta_{1,\text{vis}} \sin\phi_{1,\text{vis}} + p_{2,\text{mis}} \sin\theta_{2,\text{vis}} \sin\phi_{2,\text{vis}}
\end{aligned} \tag{4}$$

The main weakness of this method is apparent from this system of equations. Namely, when $\phi_{1,\text{vis}} \approx \phi_{2,\text{vis}} \pm \pi$ (the two visible taus are back to back in the transverse plane), the two equations (4) become degenerate, and the collinear mass performs very poorly; the collinear mass cannot be calculated at all in some cases. As well, this method is heavily dependent on $\not{E}_T$ resolution.

3.2 Missing Mass Calculator

The missing mass calculator takes a different approach to computing an invariant mass. The pseudo-rapidity, η, is defined by $\eta = -\ln\tan\frac{\theta}{2}$. The angular distance between the visible and invisible tau decay products is defined by $\Delta R = \sqrt{(\eta_{\text{vis}} - \eta_{\text{mis}})^2 + (\phi_{\text{vis}} - \phi_{\text{mis}})^2}$. $Z \to \tau\tau$ events are then simulated using Pythia and TAUOLA. These events are used to populate distributions of ΔR for each decay mode, in bins of tau transverse momentum. Each ΔR distribution is fitted with a combination of a Gaussian and Landau function. The fit parameter values are then also parameterized as a function of tau transverse momentum. Thus, a probability distribution function $P(\Delta R, p_T(\tau))$ of a given tau decay topology has been constructed.

Using this probability distribution function for a given tau decay, the probability for an overall event is then:

$$P_{\text{event}}\Big(\Delta R_1, p_T(\tau_1), \Delta R_2, p_T(\tau_2)\Big) = P\Big(\Delta R_1, p_T(\tau_1)\Big) \times P\Big(\Delta R_2, p_T(\tau_2)\Big) \tag{5}$$

For events in which both taus decay hadronically, both $M_{\tau\tau}$ and P_{event} can be calculated for each point in $(\phi_{1,\text{mis}}, \phi_{2,\text{mis}})$ phase space (since there are two more unknowns than constraints). This phase space is scanned over a grid, calculating both $M_{\tau\tau}$ and P_{event} at each point. A mass distribution is then created, weighting each point according to its probability. The overall mass output is then taken to be the most probable mass calculated from this distribution. For events in which one or both of the taus decay leptonically, the phase space scanned over must be higher dimensional. The above procedure is used in these cases, but in the $(\phi_{1,\text{mis}}, \phi_{2,\text{mis}}, m_{1,\text{mis}})$ and $(\phi_{1,\text{mis}}, \phi_{2,\text{mis}}, m_{1,\text{mis}}, m_{2,\text{mis}})$ spaces for $H \to \tau_{\text{lep}}\tau_{\text{had}}$ and $H \to \tau_{\text{lep}}\tau_{\text{lep}}$ events respectively. Similarly to the collinear mass, the MMC is sometimes not able to solve the equations (2) in order to compute a mass. This is, however, far more rare for the MMC as this would only be an issue if the system were not solvable at any of the grid points. [3]

4 Boosted Regression Trees

A boosted regression tree (BRT) is a machine learning based regression algorithm. A trained BRT takes as input a given set of kinematic variables associated with an event, and outputs an estimate of a certain target variable. In order to do this, the algorithm is first given a training dataset in order learn the relationships between the variables and the target variable. The regressor is then evaluated on a statistically independent testing dataset in order to ensure the algorithm has been trained such that it uses overall behaviour of the data, rather than statistical fluctuations. The algorithms described below are implemented with the TMVA software package [4].

4.1 Regression Trees

A boosted regression tree is based on the concept of a single regression tree. A regression tree essentially splits the phase space into regions of similar target variable values. Then, a single target variable estimate is assigned to each region, based on the mean target value in the region for the training dataset. Thus, by determining which of these regions of phase space a given event falls into, a target value estimate can be assigned. The phase space is divided by a series of binary cuts, resulting in a tree-like structure as shown in figure 1.

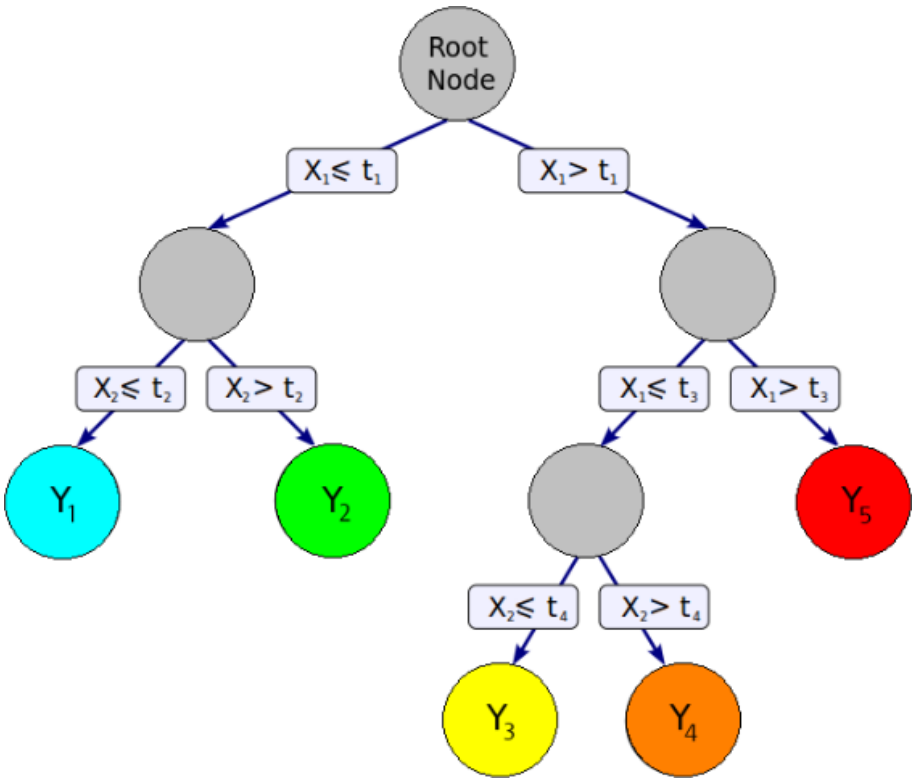

Figure 1: A graphical representation of the regression tree structure. In this case X_i represent input variables, t_i represent cut values on these variables, and Y_i represent the mean target value in each 'leaf'. [5]

In order to train a regression tree, the necessary inputs are: a training sample containing input variable values and a target variable value for each event; and regression tree training parameter values such as the maximum tree depth, splitting criteria, and number of possible splitting points to scan for each variable. The tree then begins training by scanning through the potential splitting points on each of the variables, evaluating the value of the splitting criteria. The splitting criteria used in this study is known as regression variance, or average squared error.

In order to define the splitting criteria, first define $Y(k)$ to be the target variable value for event k. The input variables for an event, k, can be denoted as $X_i(k)$ where i runs from 1 up to the total number of input variables. $P_{i,j}$ can then be defined to be the j'th potential splitting point along variable X_i. In this case, j would run from 1 up to the total number of possible splitting points along each variable. Let $N_{(R)\ i,j}$ and $N_{(L)\ i,j}$ be the number of events, k, with $x_i(k) > P_{i,j}$ and $x_i(k) \leq P_{i,j}$ respectively. Here, the (R) and (L) subscripts denote events with X_i above and below (or right and left of) $P_{i,j}$, respectively. The total number of events, $N_{(\text{tot})}$ is then given by $N_{(\text{tot})} = N_{(R)\ i,j} + N_{(L)\ i,j}$ for any i and j. Let $\langle Y \rangle_{(\text{tot})}$ be the average target value for all events. The mean target values for events with $x_i(k) > P_{i,j}$ and for events with $x_i(k) \leq P_{i,j}$, are then defined by:

$$\begin{aligned} \langle Y \rangle_{(R)\ i,j} &= \frac{1}{N_{(R)\ i,j}} \sum_{\{k \mid X_i(k) > P_{i,j}\}} Y(k) \\ \langle Y \rangle_{(L) i,j} &= \frac{1}{N_{(L)\ i,j}} \sum_{\{k \mid X_i(k) \leq P_{i,j}\}} Y(k) \end{aligned} \tag{6}$$

The variance of target values for events on either side of $P_{i,j}$ is then given by[1]:

$$\begin{aligned}\sigma^2_{(R)}(P_{i,j}) &= \frac{1}{N_{(R)\ i,j}} \sum_{\{k \mid X_i(k) > P_{i,j}\}} \Big(Y(k) - \langle Y\rangle_{(R)\ i,j}\Big)^2 \\ \sigma^2_{(L)}(P_{i,j}) &= \frac{1}{N_{(L)\ i,j}} \sum_{\{k \mid X_i(k) \leq P_{i,j}\}} \Big(Y(k) - \langle Y\rangle_{(L)\ i,j}\Big)^2\end{aligned} \tag{7}$$

The variance of target values amongst all events is:

$$\sigma^2_{(\text{tot})} = \frac{1}{N_{(\text{tot})}} \sum_{k\ \in\ \text{Events}} \Big(Y(k) - \langle Y\rangle_{(\text{tot})}\Big)^2 \tag{8}$$

Finally, the splitting parameter for a point $P_{i,j}$ is then given by:

$$\Phi(P_{i,j}) = \frac{\sigma^2_{(\text{tot})} - \frac{1}{N_{(\text{tot})}}\Big(N_{(R)\ i,j} \cdot \sigma^2_{(R)}(P_{i,j}) + N_{(L)\ i,j} \cdot \sigma^2_{(L)}(P_{i,j})\Big)}{\sigma^2_{(\text{tot})}} \tag{9}$$

After scanning over each of the possible splitting points, the point with the maximum Φ value is selected, and the phase space is divided into two regions (nodes), defined as the regions above and below the selected variable splitting point. This procedure is then applied iteratively for each resulting node, until some termination condition is reached. Possible termination conditions for a node include a minimum number or fraction of events remaining, or a maximum tree depth being reached.

Once every node has terminated, each terminal node, or leaf, is assigned an estimator value of the target variable. The estimate is taken to be the mean target variable value of the training events in the leaf.

Now that a regression tree has been trained, it can be used to estimate the target value for other events. This is simply done by determining which leaf the event would be attributed to, and taking the assigned estimator value from that leaf as an estimate of the target value.

4.2 Boosting

A single regression tree can be quite sensitive to statistical fluctuations in the training data set. In order to counteract this, an algorithm known as boosting can be applied. Boosting involves reweighting events according to their deviation from the regressor value, and retraining the regressor using these weights. The specific algorithm used in this case in known as AdaBoost.R2. In this algorithm, a loss function, L, is first defined. There are mutiple possible definitions, but for the purposes of this study, a quadratic loss function has been used. The quadratic loss function for a regressor, i, is defined by:

$$L^{(i)}(k) = \left[\frac{\Big|y(k) - y^{(i)}(k)\Big|}{\max\limits_{k'\in\ \text{Events}}\Big(\Big|y(k') - y^{(i)}(k')\Big|\Big)}\right] \tag{10}$$

where k is a given event, $y(k)$ is the target value associated with event k, and $y^{(i)}(k)$ is the output of the regressor i, for event k. The average loss of regressor i, $\langle L\rangle^{(i)}$, is defined as:

$$\langle L\rangle^{(i)} = \sum_{k'\in\ \text{Events}} w^{(i)}(k')L^{(i)}(k') \tag{11}$$

[1]Equations (6) and (7) are for unweighted data. When events have weights (eg. after boosting), each term in each sum is weighted by the weight associated with event k.

The parameter $\beta_{(i)}$ is defined as $\beta_{(i)} = \langle L \rangle^{(i)} / (1 - \langle L \rangle^{(i)})$. The weights for regressor $i + 1$ are then given by:

$$w^{(i+1)}(k) = \tilde{N} w^{(i)}(k) \cdot \beta_{(i)}^{1-L^{(i)}(k)}, \; \tilde{N} = \frac{\sum\limits_{k' \in \text{Events}} w^{(i)}(k')}{\sum\limits_{k' \in \text{Events}} w^{(i)}(k') \cdot \beta_{(i)}^{1-L^{(i)}(k')}} \tag{12}$$

The boosting process results in a forest of regression trees. In order to obtain an overall best target estimate, a weighted median is used. The output is taken to be the smallest output of a regression tree, $y^{(i)}(k)$, such that

$$\sum_{\{t \mid y^{(t)}(k) < y^{(i)}(k)\}} \ln \frac{1}{\beta_{(t)}} \geq \frac{1}{2} \sum_{t=1}^{N_{\text{trees}}} \ln \frac{1}{\beta_{(t)}} \tag{13}$$

4.3 Testing

It is important to test a BRT on a dataset independent of the training data set. In some cases, the BRT will overfit the training data. If this is the case, the BRT outputs on the training and testing samples will differ significantly. This overfitting is known as overtraining. By validating the BRT on an independent sample, this overtraining can be detected, and appropriate training parameters can be readjusted.

4.4 Interpretation

Due to their complexity, it is easy to treat a BRT as a 'black box' without understanding how it gives the results it does. However, in reality, a BRT is very similar conceptually to the MMC. For a given visible event topology, both estimators use known kinematic distributions to estimate the most likely target value. The MMC takes into account these probabilities by using distributions of angular seperation between visible and invisible tau decay products. It uses these to assign a probability to each possible invisible decay topology, and find the overall most probable mass. The BRT, on the other hand, simply takes a given visible event topology, and assigns it a target value based on the distribution of target values for events with similar event topologies from the training dataset. The BRT is merely making use of the same information as the MMC, but without the need to put these distributions in manually; instead, the algorithm is allowed to learn these distributions on its own, using the training data set.

5 Samples

There are two main simulated data sets used in this analysis. The first set of simulated data is referred to as truth level. These are samples in which the full decay kinematics are taken directly from the generator. These samples are expected to capture most of the collision physics, but not necessarily what the detector actually observes. These are useful to train a BRT, since they contain all of the essential physics, and are relatively quick to generate. However, they cannot give an accurate estimate of the performance of the algorithm. The other simulated data set is referred to as full simulation. These have been passed through a full simulation of the ATLAS detector and are expected to model what is actually detected. These are slow to generate, but essential since they can be used to evaluate the performance of the BRT in conditions similar to what is expected in real data.

5.1 Truth Level

The truth level samples used are generated using MC@NLO and showered using Pythia. There are 1,000,000 $Z \to \tau\tau$ events. $H \to \tau\tau$ events were produced from Higgs masses of 40 GeV up to 200 GeV

$X \to \tau\tau$ Samples	Truth Level	Full Simulation
Higgs Mass Range	40 - 200 GeV	100 - 150 GeV
Higgs Mass Point Spacing	1 GeV	5 GeV
Higgs Events/Mass Point	~42,000	65000 - 150,000 (skimmed)
Z Events	~420,000	180,000 (skimmed)

Table 1: A summary of the number of events used in this study. The truth level numbers are true of only the gluon-gluon fusion production mode, while the full simulation numbers are true of both gluon-gluon fusion and vector boson fusion production modes.

in 1 GeV steps. There 100,000 total events at each mass point. All currently used truth samples are produced via the gluon-gluon fusion production mechanism. In the future, samples will likely be produced via the vector boson fusion production mechanism as well. The current study investigates only events in which both taus decay hadronically. This requirement reduces event counts to around 42% of the total event counts.

5.2 Full Simulation

The full simulation samples used are the official ATLAS $Z \to \tau_{\text{had}}\tau_{\text{had}}$ and $H \to \tau_{\text{had}}\tau_{\text{had}}$ samples. The samples have been skimmed. For both vector boson fusion and gluon-gluon fusion production modes, there are about 180,000 $Z \to \tau_{\text{had}}\tau_{\text{had}}$ events after this skimming. For $H \to \tau_{\text{had}}\tau_{\text{had}}$, samples are spaced in 5 GeV increments between Higgs masses of 100 GeV and 150 GeV, for each vector boson fusion, and gluon-gluon fusion. There are between 65,000 and 150,000 at each mass point. The numbers of samples for both truth level and full simulation are summarised in table 1. For each mass point, the vector boson fusion and gluon-gluon fusion samples are weighted according to their relative cross sections and combined to form a mixed production mode Higgs sample. When the training is done on truth level events, all of these events are available for testing.

6 Truth Level Studies

In order to study the behaviour of BRT's, a study was first performed at truth level. The BRT input variables were the transverse momentum, pseudorapidity, azimuthal angle, and energy of the full tau. Since this BRT is trained on information that is not measured in data, it cannot be directly compared to the collinear or MMC mass. However, it allows us to study the behaviour of the BRT in predicting a known function (ie. $m^2c^4 = E^2 - p^2c^2$). The BRT was found to be quite biased, even in this relatively simple case. The BRT output as a function of truth mass is shown in figure 2. A slope of 1 would represent perfect agreement in the mean of the output distributions with the true mass. However, this is clearly not seen to be the case. An example fit line is shown on the plot, along with its slope, which is significantly less than 1. The graph is also quite clearly non-linear, especially near the edges of the mass range.

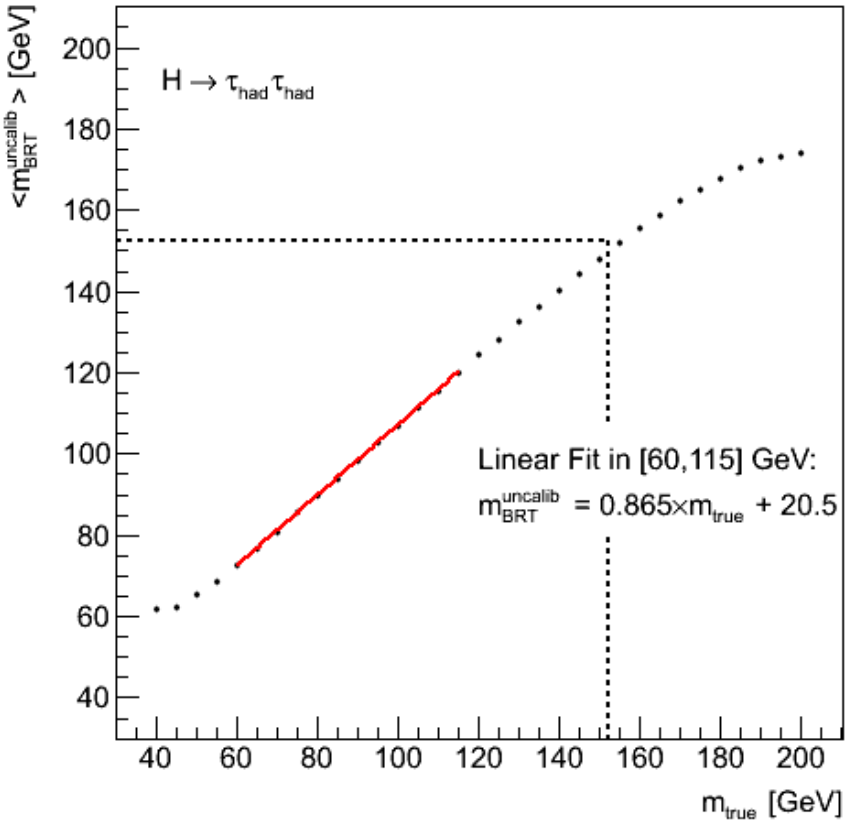

Figure 2: Reconstructed BRT mass as a function of truth mass for truth level samples. The output is clearly biased towards the middle of the mass range, having a slope of less than 1.

These results were obtained using the following training parameters: 40 Trees, a minimum of 20 events per node, a maximum tree depth of 100, 20 possible splitting points per variable, and an Adaboost learning rate of 0.2. Experimenting with training parameters appeared to be able to reduce, but not remove this bias. It thus appears that these are characteristics of the BRT itself, and through parameter tuning, these effects can be minimized.

A BRT can also be trained and tested at truth level, but missing the information due to the unmeasured neutrinos. In this case, the only observable associated with the neutrinos is taken to be the vector sum of the neutrino momenta projected into the transverse plane. This vector is taken to be equivalent to the missing transverse energy in the detector. A list of variables was then created, derived from the measured tau momenta, and this truth level missing transverse energy. A BRT is then trained with this variable list. As expected the performance degrades relative to the previous study, due to the missing information. The bias, however is still present.

7 BRT Mass Estimator

7.1 Training

A BRT was trained using the invariant mass of the two tau system as the target output. The input variables used are listed in table 7.1. The current variable list is fairly preliminary, and could likely be

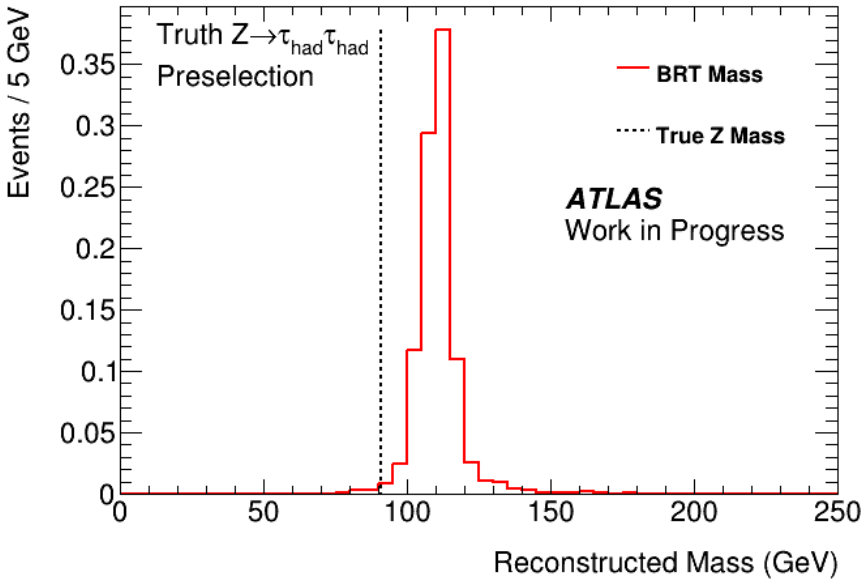

Figure 3: Reconstructed BRT mass distribution for truth level $Z \rightarrow \tau\tau$ samples. The distribution is clearly shifted from the correct mass value.

improved upon and reduced. The training parameters used were determined through a scan over possible parameter values. The BRT was trained using 80 trees. There were 20 possible cut values allowed along each variable. Each tree was allowed to grow until either a maximum depth of 100, or a minimum of 40 events remaining in a node.

As a first check, the BRT can be tested on truth level $Z \rightarrow \tau\tau$ samples. The output mass distribution is shown in figure 3. It shifted by about 20 GeV from the correct Z mass value. As a first attempt to account for this, a calibration can be applied. The simplest such calibration is a simple constant shift such that the truth level Z peak reconstructs correctly. For all testing on full simulation samples, this calibration has been applied. The calibration is needed due to the bias discussed in section 6. More complex calibrations were also attempted in order to correct for non-constant shifts over the mass range, but any such calibration was found to widen the distributions such that the calibration worsened the overall performance.

7.2 Full Simulation Testing

In order to allow for the best possible training statistics, no truth level Higgs events were put aside for testing. Instead, all evaluation is done on full simulation samples. The testing is done on the full simulation samples described in section 5. In order to test the applicability of the BRT to the current ATLAS $H \rightarrow \tau\tau$ analysis, the BRT is tested using the preselection cuts applied in the analysis [6]. As well, it is tested seperately in each of the categories used in the analysis. The 'Boosted' category is intended to capture events from the gluon-gluon fusion production mode. Thus, since training is done on gluon-gluon fusion samples, this category should give the best comparison between the mass estimators. Ideally, a seperate BRT would be trained on vector boson fusion produced samples in order to be used in the 'VBF' category. Focus will be placed on the Boosted category for this analysis.

The first performance test is a simple check of the output mass distributions for full simulation $Z \rightarrow \tau\tau$ samples. The distributions for each of the mass estimators is shown in figure 4. The collinear mass appears to have largest width. By calulating the root-mean-square distance from the mean and

Variable	Description
tau1_pt	The transverse momentum of the leading visible tau in the event.
tau1_eta	The pseudorapidity of the leading visible tau in the event.
tau2_pt	The transverse momentum of the subleading visible tau in the event.
tau2_eta	The pseudorapidity of the subleading visible tau in the event.
met_et	The missing transverse energy in the event.
visible_mass	The invariant mass of the visible decay products of the two taus.
dphi_tau1_met	The azimuthal angular distance between the leading visible tau, and the missing transverse energy.
dphi_tau1_met	The azimuthal angular distance between the subleading visible tau, ,and the missing transverse energy.
dphi_tau_tau	The azimuthal angular distance between the visible decay products of the two taus.
deta_tau_tau	The difference in pseudorapidity between the visible decay products of the two taus.
dR_tau_tau	The angular distance between the visible decay products of the two taus.
ptsum_tau_tau	The magnitude of the vector sum of the transverse momenta of the two visible taus.
ptsum_tau_tau_met	The magnitude of the vector sum of the transverse momenta of the two visible taus and the missing transverse energy.
pttot_tau_tau	The scalar sum of the transverse momenta of the two visible taus, divided by ptsum_tau_tau.
pttot_tau_tau_met	The scalar sum of the transverse momenta of the two visible taus and the missing transverse energy, divided by ptsum_tau_tau_met.
ptdiff_tau_tau	The scalar difference in transverse momentum between the two visible taus, divided by ptsum_tau_tau.
transverse_mass_tau_tau	The transverse mass of the two visible taus in the event.
transverse_mass_tau1_met	The transverse mass of the leading visible tau with the missing transverse energy.
transverse_mass_tau2_met	The transverse mass of the subleading visible tau with the missing transverse energy.

Table 2: A table of the variables used in training the BRT for invariant mass estimation.

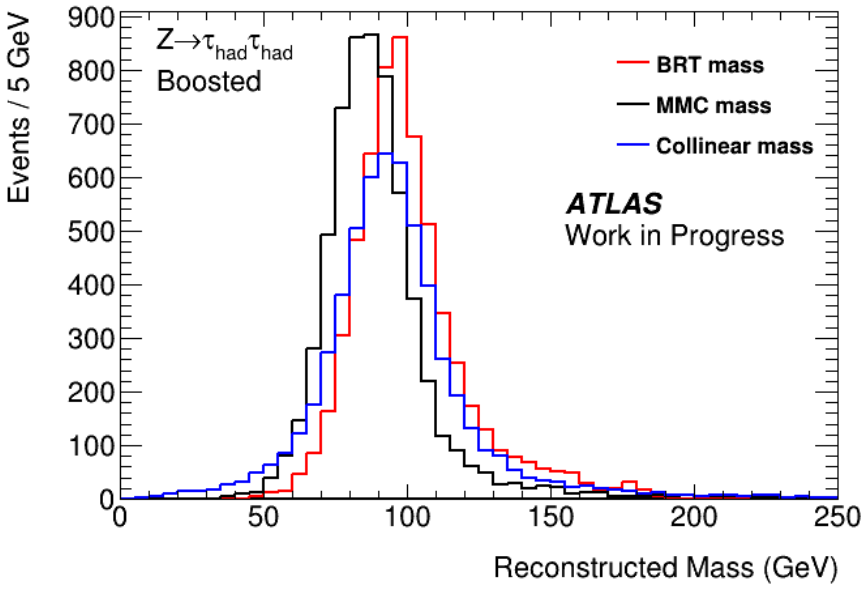

Figure 4: Reconstructed mass for full simulation Z boson samples. The BRT performs quite well, but has a significant high mass tail.

dividing by the true mass value (91.1876 GeV), we obtain a resolution estimate. This value is found to be 29.6% for the collinear mass. The BRT improves upon this with an estimated resolution of 22.5%. At first glance, this appears to match the performance of the MMC which also has an estimated resolution of 22.5%. However, this metric is fairly limited, as it fails to account for the biases visible in the BRT output. As well, the current BRT has a high mass tail that may be problematic for its distinguishing power between the Z and Higgs masses. The mean of the BRT output distribution overestimates the true Z mass by about 9 GeV. Similarly, the collinear mass overestimates it by about 5 GeV. The MMC, however, is the most accurate, underestimating it by only about 1 GeV.

The same check can be performed at each of the Higgs mass points. The output distribution for a Higgs mass of 125 GeV is shown in figure 5. The resolutions at this point are significantly better than for the Z samples for each of the estimators. This is due to the narrow decay width of the Higgs, relative to the Z boson. The MMC and BRT each give an estimated resolution of 13.7%. For comparison, the collinear mass has a resolution of about 21.2%. However, once again, this metric fails to take into account biased mass estimator output. At this mass, the collinear and BRT masses have the most accurate means at this mass, with the collinear underestimating the true mass by about 1 GeV, and the BRT overestimating it by about the same amount. The MMC, on the other hand, underestimates the true value by about 9 GeV.

The information from these distributions can then be used to plot reconstructed mass versus truth mass for each of the three estimators. The collinear mass has the least bias, with each distribution peaking at almost exactly the correct mass value. However, the distributions are very wide. On the other hand, the MMC has a constant shift, as well as a slope that is slightly less than one, but with much narrower distributions than the collinear mass. Similarly to the MMC, the BRT has a (non-constant) slope of less than one, meaining that it is offset from the true value at some mass points. The width of the distributions for the MMC an BRT are very comparable.

There are many possible metrics to quantify the performance of the different estimators. The resolutions listed above, for example, give an indication of performance, but lack information about the bias of the outputs of the various estimators over the mass range. As well, information about the shape of the distributions is missing. Possibly a more effective metric to rank the classifiers would be to treat the

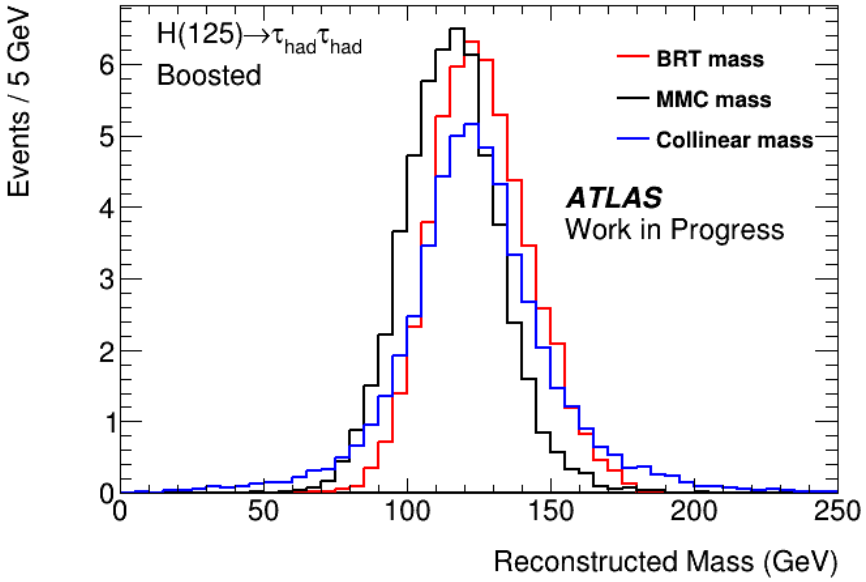

Figure 5: Reconstructed mass for full simulation 125 GeV Higgs boson samples. The BRT performs well, with comparable width to the MMC.

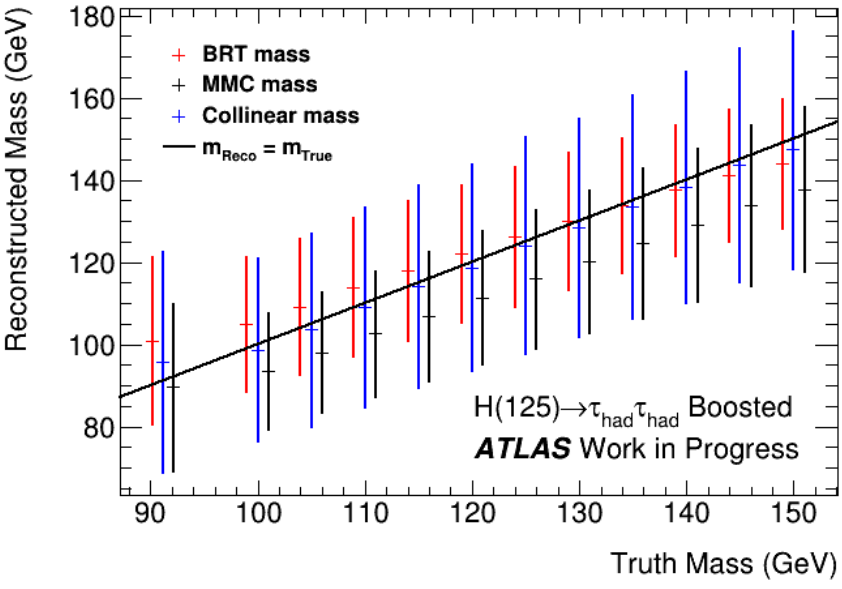

Figure 6: Reconstructed mass as a function of truth mass for full simulation samples. The BRT is biased, but is competitive with the MMC.

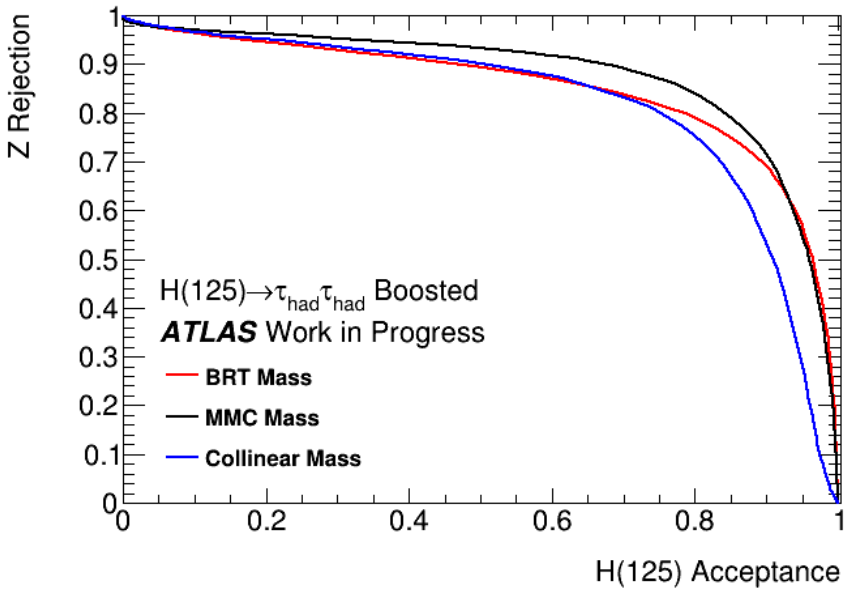

Figure 7: Reconstructed BRT mass as a function of truth mass for truth level samples. The output is clearly biased towards the middle of the mass range, having a slope of less than 1.

Z boson samples as a 'background' distribution, and 125 GeV Higgs boson samples as a 'signal' distribution, and create a receiver operating characteristic (ROC) curve for the two. This curve is created by scanning through mass values for each estimator, and plotting the fraction of Z events below that value versus the fraction of 125 GeV Higgs events above the value. Such a curve is shown in figure 7. The BRT clearly outperforms the collinear mass, while the MMC outperforms both other estimators. The two curves are very comparable at large values of 125 GeV acceptance. The lack of performance at low values is likely due to the large high mass tail of the BRT output Z boson mass distribution.

8 Timing

The biggest advantage of the BRT over the MMC is the dramatic decrease in computation time required. The computation time was measured by first measuring the time to complete a loop through all the events, without calculating either the MMC or BRT, or any of the BRT input variables. The measurement was then repeated with the additional calculation of the BRT input variables and BRT output for each event. The time was measured a third time, but calculating the MMC mass, again without the BRT mass or input variables. The BRT and MMC times are then taken to be differences in time between their respective event loops, and the empty event loop. The initialization time for both the MMC and BRT were not included in the timing, since these do not scale with the number of events and should be negligible for the purposes of any analysis dealing with large amounts of data. The time taken per 1000 events for the MMC is about 369.2 s. For the BRT it is about 0.2 s. This is an improvement by a factor of about 1500. This is made even more significant by the fact that the mass for each event must be recalculated for each systematic uncertainty unvolved in the analysis.

9 Future Improvements

There are many possible ways to attempt to improve the current BRT. First, the problem of reducing bias and variance in regressors in machine learning has been extensively studied [7] [8]. By utilizing the techniques suggested in such studies, the BRT training parameters could be better optimized. As well, ideally training would be done on full simulation samples, such that there would be better agreement between the distributions for training and testing. The computational cost of this prevents it from being immediately feasible for the full mass range of truth samples currently used for training. Finally, other machine learning based regression algorithms could be explored. Other algorithms may be better suited to this problem than a BRT.

10 Summary and Conclusion

A boosted regression tree has been shown to give competitive performance to the currently used Missing Mass calculator in predicting the invariant mass of hadronically decaying tau lepton pairs. With future work, the BRT could reach, or even surpass the MMC's performance. As well, taking just a tiny fraction of the computation time, this new technique could greatly reduce the time needed for an analysis.

11 Acknowledgements

Thank you to Dr. Dugan O'Neil and Dr. Quentin Buat for all of their assistance with this project. As well, I would like to thank Natascha Hedrich and Dr. Andres Tanasijczuk for their contributions to this work[9].

References

[1] G. Aad, T. Abajyan, B. Abbott, J. Abdallah, S. Abdel Khalek, A. Abdelalim, O. Abdinov, R. Aben, B. Abi, M. Abolins, et al., *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, Physics Letters B **716** no. 1, (2012) 1–29.

[2] S. Chatrchyan, V. Khachatryan, A. M. Sirunyan, A. Tumasyan, W. Adam, E. Aguilo, T. Bergauer, M. Dragicevic, J. Erö, C. Fabjan, et al., *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC*, Physics Letters B **716** no. 1, (2012) 30–61.

[3] A. Elagin, P. Murat, A. Pranko, and A. Safonov, *A new mass reconstruction technique for resonances decaying to $\tau\tau$*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **654** no. 1, (2011) 481–489.

[4] A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, and H. Voss, *TMVA: Toolkit for Multivariate Data Analysis*, PoS **ACAT** (2007) 040, `arXiv:physics/0703039`.

[5] K. van Nieuwkoop, *Bringing the Higgs Boson to Rest*, Master's thesis, Simon Fraser University, 8888 University Dr, Burnaby, BC V5A 1S6, Canada, 2013.

[6] A. collaboration et al., *Evidence for Higgs boson decays to the τ+ τ- final state with the ATLAS detector*, ATLAS-CONF-2013-108. 2013.

[7] Y. L. Suen, P. Melville, and R. J. Mooney, *Combining bias and variance reduction techniques for regression trees*, pp. , 741–749. Springer, 2005.

[8] L. Breiman, *Using iterated bagging to debias regressions*, Machine Learning **45** no. 3, (2001) 261–277.

[9] N. S. Hedrich, *Boosted Regression Trees in the H→ ττ decay channel*, Aug, 2013.