SLAC-TN-12-015

Extracting $b\bar{b}$ Higgs decay signals using multivariate techniques

W Clarke Smith

Office of Science, Science Undergraduate Laboratory Internship (SULI)

George Washington University

SLAC National Accelerator Laboratory

Menlo Park, CA

August 17, 2011

Prepared in partial fulfillment of the requirements of the Office of Science, Department of Energy's Science Undergraduate Laboratory Internship under the direction of Timothy Barklow with the ATLAS group at SLAC National Accelerator Laboratory.

Participant: _____

Signature

Research Advisor: _____

Signature

Work supported by US Department of Energy contract DE-AC02-76SF00515.

TABLE OF CONTENTS

ABSTRACT

Extracting $b\bar{b}$ Higgs decay signals using multivariate techniques. W CLARKE SMITH (George Washington University, Washington, DC 20052) TIMOTHY BARKLOW (ATLAS group at SLAC National Accelerator Laboratory, Menlo Park, CA 94025)

For low-mass Higgs boson production at ATLAS at $\sqrt{s} = 7$ TeV, the hard subprocess $gg \to h^0 \to b\bar{b}$ dominates but is in turn drowned out by background. We seek to exploit the intrinsic few-MeV mass width of the Higgs boson to observe it above the background in $b\bar{b}$-dijet mass plots. The mass resolution of existing mass-reconstruction algorithms is insufficient for this purpose due to jet combinatorics, that is, the algorithms cannot identify every jet that results from $b\bar{b}$ Higgs decay. We combine these algorithms using the neural net (NN) and boosted regression tree (BDT) multivariate methods in attempt to improve the mass resolution. Events involving $gg \to h^0 \to b\bar{b}$ are generated using Monte Carlo methods with PYTHIA and then the Toolkit for Multivariate Analysis (TMVA) is used to train and test NNs and BDTs. For a 120 GeV Standard Model Higgs boson, the m_{h^0}-reconstruction width is reduced from 8.6 to 6.5 GeV. Most importantly, however, the methods used here allow for more advanced m_{h^0}-reconstructions to be created in the future using multivariate methods.

INTRODUCTION

When formulating the Standard Model of particle physics (SM), theorists initially had trouble explaining the origin of mass for bosons and fermions. In general, they reasoned that a symmetric system could not evolve into an asymmetric state. As a consequence, though, neither $W^{\pm}$ nor Z^0 bosons would be massive. Since $W^{\pm}$ and Z^0 are massive, they hypothesized the existence of a so-called *Higgs mechanism* that allows for spontaneous symmetry breaking, thereby giving such force-carrying bosons mass.

A Toroidal LHC ApparatuS (ATLAS), one of the six experiments at the Large Hadron Collider (LHC), is currently attempting to confirm the empirical existence of the Higgs mechanism by observing its scalar remnant: h^0, the *Higgs boson*[1]. Teams within ATLAS are attempting to do this by detecting the particles that characteristically result when these bosons decay. Higgs bosons can be created in proton-proton collisions, or 'events,' at ~ 7 TeV in the center-of-mass frame. The most promising Higgs decay modes to detect are $WW^{(*)}$[2], $ZZ^{(*)}$, $\tau^+\tau^-$, and $\gamma\gamma$ [1]. However, for the low Higgs masses ($115 \lessapprox m_h \lessapprox 140$ GeV) that this project is interested in, the Higgs boson decays into $b\bar{b}$ approximately 66% of the time[3] via the hard subprocess $gg \to h \to b\bar{b}$. We call *signal events* those that include this process (Figure 1) and *background events* those that include a different $b\bar{b}$-producing hard subprocess, such as $gg \to b\bar{b}$ (Figure 2). Unfortunately, almost every $b\bar{b}$-dijet detected results from a background event.

In order to successfully detect the Higgs boson, we seek to exploit its intrinsic narrow mass width of several MeV [2]. Theoretically, this will manifest itself as a sharp, narrow spike at m_h in the $b\bar{b}$-dijet mass ($m_{b\bar{b}}$) plot, allowing for the Higgs boson to be easily observed. However, reconstructing $m_{b\bar{b}}$ from event information proves difficult due to the so-called 'jet

[1]There may be multiple Higgs bosons; we reduce to the singular case. Also, hereafter we denote the Higgs boson h instead of the more accurate h^0 (the symbol for a scalar, light Higgs boson).

[2]The superscript $(*)$ denotes a virtual particle.

[3]Calculated for a 120 GeV SM Higgs boson using PYTHIA Monte Carlo generated events.

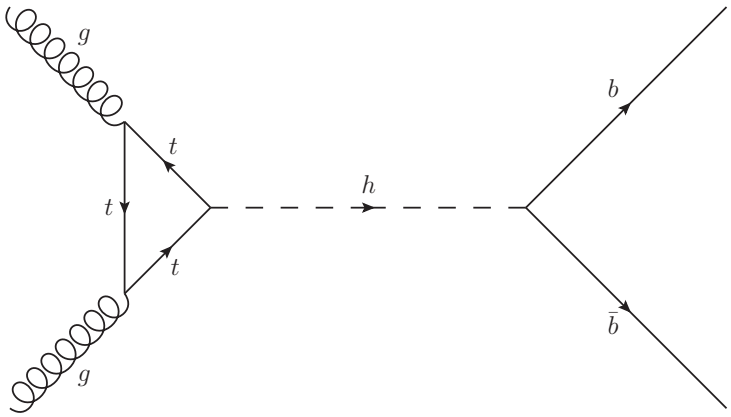


Figure 1: Feynman diagram of $gg \to h \to b\bar{b}$ (Higgs boson production from gluon fusion via a virtual t-quark loop followed by $b\bar{b}$ Higgs decay), the hard subprocess in signal events.

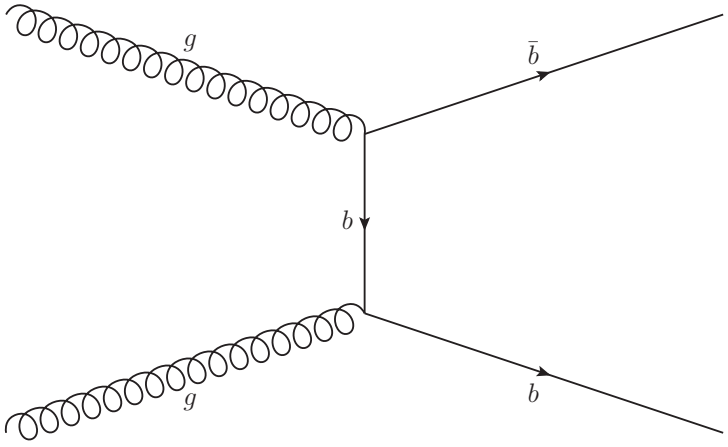


Figure 2: Feynman diagram of $gg \to b\bar{b}$ (production of $b\bar{b}$ from t-channel gluon fusion via the exchange of a virtual b-quark), an example of a hard subprocess in background events.

combinatorics problem,' that is, our inability to identify jets accurately. Although the $b\bar{b}$-dijets can be efficiently distinguished (b-tagged) [3], the b- and $\bar{b}$-quarks constantly radiate and reabsorb gluons before hadronizing and some gluons will be radiated but not reabsorbed. Those gluons are called *final state radiation* (FSR) and they are precisely those whose jets are difficult to identify. To solve this problem, there have been many *mass-reconstruction*

algorithms developed to isolate such FSR-jets, but their inability leads to a reconstructed $m_{b\bar{b}}$ plot with a dull, essentially nonexistent bump at m_h. Realizing that the plot of background events vs. $b\bar{b}$-dijet masses experiences gradual decay over tens of GeV and so should the analogous reconstructed background $m_{b\bar{b}}$ plots, we restrict ourselves to the case of signal events, where $m_{b\bar{b}} = m_h$. In other words, reconstructing $m_{b\bar{b}}$ from signal events alone is sufficient to extract the m_h spike from the background $m_{b\bar{b}}$ plot. Hence, the goal of this project is to specifically construct an improved m_h-reconstruction algorithm[4].

Motivated by the quantity and unknown degree of interdependency of the observables measured by ATLAS, we use several types of *multivariate analysis*[5] (MVA) to search for patterns in Monte Carlo (MC) generated pp collisions in attempt to reconstruct $m_{b\bar{b}}$ more effectively. One MVA method that we use is the *neural network* (NN), a composition of many linear and nonlinear functions which maps sets of input values into models of the target values [4]. For our purposes, the target is $m_{h,true}$.

This approach is not altogether new; others have used various MVA techniques to enhance Higgs decay signals in many of the modes. For example, Hakl et al. [5] used NNs[6] and Takahashi [1] used another type of MVA, boosted decision trees, to improve the signal-to-background ratio for $b\bar{b}$ and $WW^{(*)}$ Higgs decays, respectively. In fact, boosted decision trees have shown impressive results in comparison to NNs — in a b-tagging performance evaluation, boosted decision trees demonstrated an efficiency improvement of 35% [6]. Instead of using a classification NN or boosted decision trees to distinguish signal events from background events like Hakl et al. and Takahashi, we use a variety of regression NNs and boosted regression trees to reconstruct m_h.

[4]Note that "mass-reconstruction algorithm" is taken to mean "$m_{b\bar{b}}$-reconstruction algorithm."
[5]Multivariate analysis is merely drawing inferences from data sets with multiple statistical variables.
[6]Hakl et al. used NNs with switching units optimized via a genetic algorithm.

MATERIALS AND METHODS

All regression MVA methods search for patterns in statistical variables in an attempt to better model the target, in this case, the true Higgs mass. If the method is not supplied any information aside from the variables, it will be entirely unable to model the target and it will not be able to produce an output; it needs to be "taught." We systematically feed the method variables as well as targets. It then develops a mechanism for predicting the target (computing outputs). This is called *training*.

Event generation with PYTHIA

For the data from which to select the variables and targets, we generate signal pp collisions using MC methods with PYTHIA. The PYTHIA program is a powerful particle physics event generator that includes a collection of methods for modeling the evolution of few-body hard processes to complex multihadronic final states. In PYTHIA event generation, the user is allowed to select the physical models used, the hard subprocesses involved, and, if applicable, the Higgs mass [7]. We use SM physics, the process $gg \to h \to b\bar{b}$, and $m_h = 90$, 100, 110, 120, 130, 140, and 150 GeV to generate 10^5 (signal) events at each m_h. PYTHIA's main weakness is that it does not include next-to-leading-order perturbative calculations and thus is insensitive to events with a low total *transverse momenta* p_T, that is, the momentum perpendicular to the beam axis. The high total p_T of $gg \to h \to b\bar{b}$ makes PYTHIA ideal for our low mass Higgs decays [7]. We do not use data taken by ATLAS. The reasons for this are twofold: we avoid complications caused by experimental uncertainties and we have absolute control over the type of event that is generated. For instance, we can choose whether to generate a $b\bar{b}$ or $ZZ^{(*)}$ Higgs decay, an event with $m_h = 90$ GeV or $m_h = 150$ GeV, or an event with one FSR-jet or five. Moreover, the only detector simulation is an angular cut is applied to remove all hadrons that escape the detector geometry; even neutrinos are

generated.

Event processing with ROOT

For each hadron detected by ATLAS, the azimuthal angle ϕ, the *pseudorapidity* $\eta = -\ln\tan\left(\frac{\theta}{2}\right)$, and p_T are measured, providing us with its geometric and energetic image[7]. Additionally, we form the observable $\sum_i p_{T,i}$ for particles i in a given event to gauge the total transverse momentum.

Using a given mass-reconstruction algorithm, first the detected hadrons are partitioned into their respective jets. This partitioning is somewhat trivial — there is not much variation between algorithms. For each jet, η, ϕ, and p_T are computed from the η_i, ϕ_i, and $p_{T,i}$ measured for each hadron i within that jet. Second, the $b\bar{b}$-dijets and FSR-jets all resulting from $b\bar{b}$ production are isolated. This is done with cuts in the $\eta\phi$ plane, where the metric $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ can be constructed, providing a notion of distance between components of the event. Each mass-reconstruction algorithm uses a value of ΔR to classify all jets within a certain vicinity of the tagged $b\bar{b}$-dijets as FSR-jets from the b- and $\bar{b}$-quarks. From this jet selection, $m_{b\bar{b}}$, $p_{T,b\bar{b}}$, and $\eta_{b\bar{b}}$ are calculated. Finally, the algorithms compute $\Delta R_{b\bar{b}}$, the "distance" between b- and $\bar{b}$-jets, from the algorithmically determined $b\bar{b}$-dijet information.

It then seems reasonable that some algorithms would be better for certain types of events than others, depending primarily on their method of selecting those jets that result from $b\bar{b}$ production. For example, some mass-reconstruction algorithms may be very good at determining $m_{b\bar{b}}$ in events with a large fraction of jets being emitted due to *initial state radiation*[8] (ISR), while others may be better suited for events with little ISR activity.

We use ROOT, the leading data analysis framework for particle physics, to process the raw output of the Pythia signal event generation. For every Pythia MC generated event,

[7]Here, θ is the angle with respect to the beam axis.

[8]Initial state radiation is those gluons that radiate but do not get reabsorbed prior to the pp collision.

m_h, $p_{T,h}$, and η_h are reconstructed using 25 of the most effective mass-reconstruction algorithms. Furthermore, these 25 algorithms are divided into five families. The algorithm families have slightly different methods for partitioning hadrons into jets while algorithms within a family have distinct ΔR values for identifying those FSR-jets that result from $b\bar{b}$ production. So, for the same events, η_b, $\eta_{\bar{b}}$, $p_{T,b}$, $p_{T,\bar{b}}$, and $\Delta R_{b\bar{b}}$ are reconstructed using the five algorithm families. Note that no ϕ information is reconstructed; this is due to the fact that the pp collisions are azimuthally symmetric. Together, these collections of outputted event information are the input values for the MVA methods. We call these input types *variables*; they are listed in Table 1. Obviously, not all 101 variables will be used for every MVA technique.

Input Variables	*Quantity per Event*	*Description*
$\sum_i p_{T,i}$	1	Total event transverse momentum
η_b, $\eta_{\bar{b}}$	5 each	Reconstructed b, $\bar{b}$-jet pseudorapidities
$p_{T,b}$, $p_{T,\bar{b}}$	5 each	Reconstructed b, $\bar{b}$-jet transverse momenta
η_h	25	Reconstructed Higgs pseudorapidity
$p_{T,h}$	25	Reconstructed Higgs transverse momentum
$m_{h,reconstruction}$	25	Reconstructed Higgs mass
$\Delta R_{b\bar{b}}$	5	Reconstructed "distance" between b, $\bar{b}$-jets

Table 1: A comprehensive list of MVA input variables.

Finally, in ROOT we declare one further characteristic of the generated events: the true Higgs mass as specified in PYTHIA. This $m_{h,true}$ is not a variable but the target of the MVA regression; it is only used in training and for performance evaluation.

MVA training and testing

At this point, we have $\sim 7 \times 10^5$ PYTHIA generated signal events each with one of seven Higgs masses as well as values for 101 different variables. The goal is then to construct mappings from the 101 variables onto the set of seven Higgs masses by invoking multiple MVA

methods, which have been shown to be superior to uni- and bivariate techniques for most purposes [1]. While each mass-reconstruction algorithm is insufficient on its own, feeding the outputs of 25 different algorithms into an advanced MVA technique vastly improves the m_h regression. Due to the ease of implementation and the customizability of its MVA methods, we use the Toolkit for Multivariate Analysis (TMVA) to construct such mappings. TMVA is included in the ROOT package and offers thirteen MVA techniques for *classification* (directly separating signal from background events), eight of which can also be used for *regression* (creating a regression function that models the target). We restrict ourselves to the two most promising of the eight methods suitable for our regression task: neural nets and boosted regression trees.

Previous NN analyses of this kind used the ROOT class `TMultiLayerPerceptron` which allows for any number of intermediate functions as well as one of six different learning methods to be used [2]. The advantages of using TMVA are the facility of using boosted regression trees, an increased number of adjustable method-specific parameters, and most importantly, the ability to repeat analyses quickly using different parameters, variables, MVA methods, and data sets [8].

In a feedforward NN[9], there is an input layer, a hidden layer(s), and an output layer (Figure 3). The nodes are called *inputs* in the input layer, *neurons* in the hidden layers, and *outputs* in the output layer. Each variable corresponds to an input and the targets correspond to the outputs. Each neuron in the first hidden layer maps a linear combination of the inputs through a specified nonlinear function σ. Neurons in subsequent hidden layers repeat this process using neurons in the previous hidden layer. Additionally, there can be "empty" neurons and inputs called *biases* which are merely weights independent of prior layers added to increase the ability of the NN.

[9]Other types of NNs (those with feedback loops, etc.) are neglected because they are overkill for our relatively standard regression task.

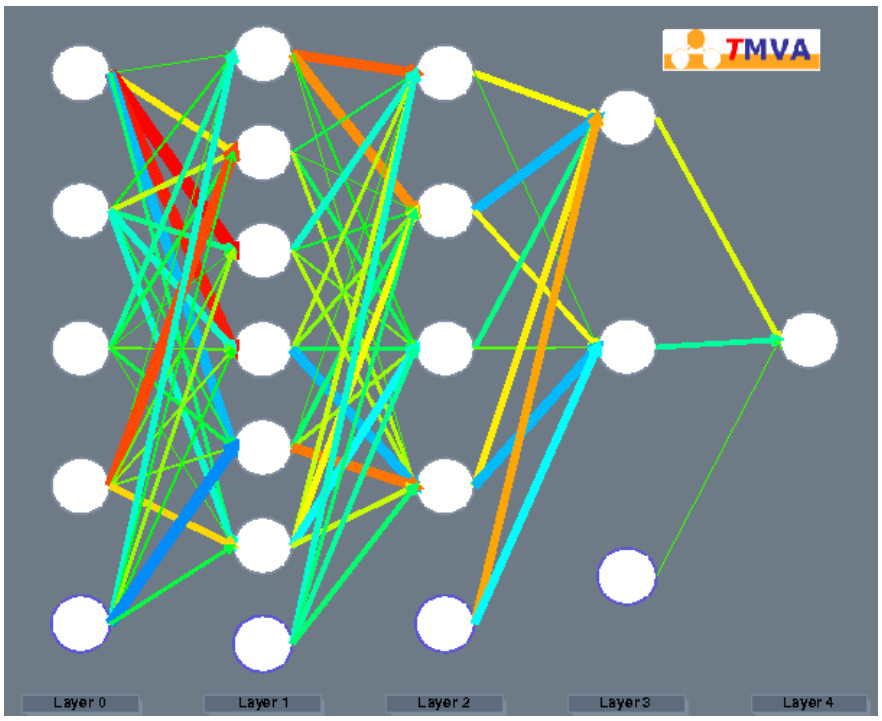


Figure 3: Feedforward NN with three hidden layers with 6, 4, and 2 neurons, respectively, as well as biases.

It may seem that having more neurons and hidden layers allows for a more complicated and therefore capable NN to be formed. This is true up to a point — while additional neurons and hidden layers enable the neural net to find more subtle relationships, it eventually reaches a peak beyond which the neural net is being *overtrained*. In that region, neural nets find patterns that are not necessarily representative of legitimate relationships, but are merely due to the finite size of the event sample. Moreover, there is not good reason to use excess hidden layers or neurons, for the Weierstrass approximation theorem dictates that any continuous function can be approximated by a sufficient number of neurons in one hidden layer [9].

MVA for particle physics has been dominated by the use of NNs over the past decade. In past several years, though, boosted decision and regression trees have emerged as a powerful tool for MVA. First of all, a *regression tree* is an MVA method grown by splitting the training sample of events into two parts for each value of each variable, depending on that

value. Then, the variable and value that produces the optimal split ("best" division of the training sample into high and low target value $m_{h,true}$ parts) is chosen as the first branching of the regression tree. This process is repeated until the user-specified maximum number of leaves is reached. *Boosted regression trees* (BDTs[10]), then, are merely an advanced form of regression trees. Boosting is achieved by allowing the tree to predict the target $m_{h,true}$ for each of the training events; those events which have their targets incorrectly predicted are given additional weight depending on how poor the prediction was. The regression tree is then regrown; this boosting process goes through a user-specified number of iterations, eventually arriving at a final boosted regression tree (Figure 4). Typically a larger number of maximum leaves and boosting iterations allows for a more robust network, but BDTs are very susceptible to overtraining so these parameters need to be chosen carefully. In attempt to minimize the effects of overtraining, once the BDT is grown and boosted, it is *pruned* by removing statistically insignificant leaves.

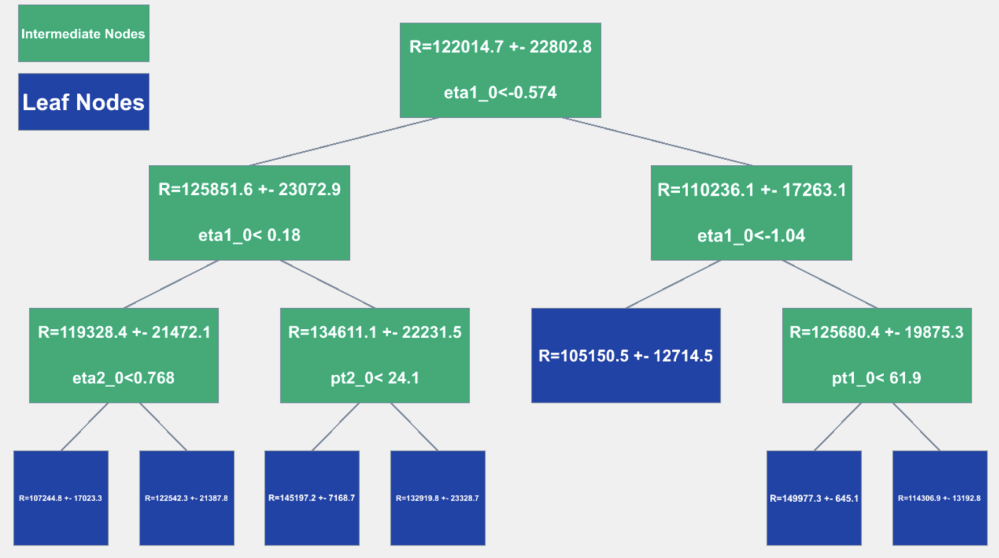


Figure 4: The 99th iteration of a boosted regression tree used for m_h regression.

[10]The acronym "BDT" is adopted for both boosted regression trees as well as boosted decision trees out of convention.

As mentioned before, event processing yields 101 variables with which to train the MVA methods. In order to increase efficiency and reduce computing time, redundant and ineffective variables are removed. Redundant variables are those which are very highly correlated with other variables used, meaning the underlying event information that they hold is very similar. Ineffective variables are those which do not show a significant correlation with the target. Clearly, redundant variables are useless while ineffective variables slow down and hinder the training process.

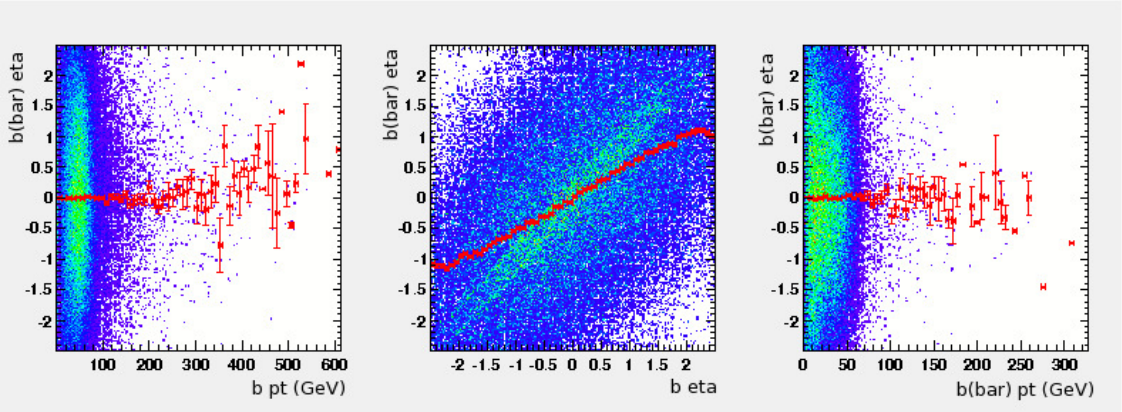
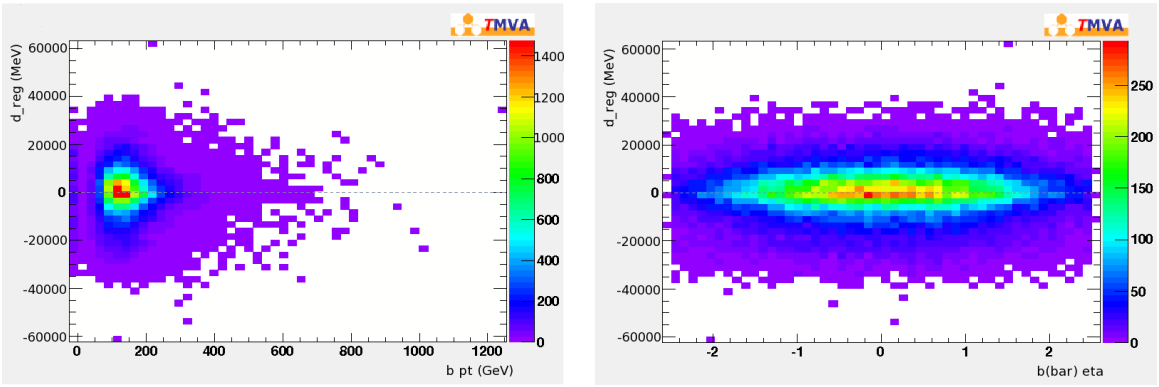


Figure 5: Plots of $\eta_{\bar{b}}$ vs. $p_{T,b}$, $\eta_{\bar{b}}$ vs. η_b, and $\eta_{\bar{b}}$ vs. $p_{T,\bar{b}}$ showing redundancy in the center plot.

A simple correlation-to-target ranking performed by TMVA allows us to reduce the number of variables from 101 to 40 relatively easily. The 25 reconstructed Higgs masses are the most highly correlated with $m_{h,true}$ and so they are not removed. Then, the correlations are similar from variable to variable, so $\sum_i p_{T,i}$ as well as two of each $\Delta R_{b\bar{b}}$, η_b, $\eta_{\bar{b}}$, $p_{T,b}$, $p_{T,\bar{b}}$, η_h, and $p_{T,h}$ are used for the initial training. Then, scatter plots of the variables vs. one another (Figure 5) as well as two-dimensional histograms of regression output deviation $d_{reg} := m_{h,reco} - m_{h,true}$ vs. the variables (Figure 6) are used to determine redundancy and effectiveness, respectively. Clearly the efficacy of a variable depends on the MVA method for which it is training. As such, we construct two sets of variables, a NN set and a BDT

set, which are slightly different but contain $\sim$30 variables.



(a) The high efficacy of $p_{T,b}$.

(b) The low efficacy of $\eta_{\bar{b}}$.

Figure 6: Regression output deviation d_{reg} vs. variable plots used to determine variable contribution to the MVA method.

Once the variables have been selected to maximize efficiency, the various parameters of the NN and the BDT need to be optimized. There are three types of NN implemented in TMVA: the Clermont-Ferrand NN, the ROOT NN (`TMultiLayerPerceptron`), and the multilayer perceptron (MLP) NN. We use the MLP NN due to its improved speed and flexibility over the other two types of neural net [8]. Five of the most important parameters for the MLP are `NCycles`, `HiddenLayers`, `NeuronType`, `NeuronInputType`, and `TrainingMethod`. `NCycles` is merely the number of training cycles the NN undergoes and `HiddenLayers` specifies the number of hidden layers and how many neurons are in each of them. In general, we want to have as many cycles, hidden layers, and neurons as possible in order to have a robust and sensitive network while not overtraining it. The `NeuronType` is the NN's nonlinear function σ. The options include the sigmoid function $1/(1+e^{-rv})$, the hyperbolic tangent function $(e^v - e^{-v})/(e^v + e^{-v})$, and the radial function $e^{-v^2/2}$. We have found the best results using the hyperbolic tangent function[11]. The option `NeuronInputType` specifies how the linear combinations are taken within each neuron. They can be sums, sums of squares,

[11] "Best" is ambiguous; here, we refer to the MLP with the lowest d_{reg} root mean square (RMS).

or sums of absolutes; we use regular sums. Finally, the `TrainingMethod` can be either the back-propagation (BP) method or the Broyden-Fletcher-Goldfarb-Shannon (BFGS) method. BFGS requires less iterations to receive the same result as BP, but it costs a large amount of computing power when the NN is complicated [8]. As such, we use BP with complex hidden structures and BFGS with simple hidden structures.

For the BDT, some of the adjusted parameters are `NTrees`, `BoostType`, `SeparationType`, `nEventsMin`, and `PruneMethod`. `NTrees` is merely the number of times the regression tree is boosted and regrown while `nEventsMin` is the minimum number of events in any given leaf node. Increasing the former and decreasing the latter heightens the BDT's sensitivity and capability so long as overtraining is avoided. For regression, the `BoostType` options are narrowed to adaptive boost (AdaBoost.R2) and gradient boost (GradientBoost). The difference between the two algorithms is the function they use to determine how poor the BDT's estimation of the target value is. Because neither AdaBoost.R2 nor GradientBoost is particularly advantageous, we use two BDT implementations: one for each type of boosting. Another important parameter is `SeparationType`, which determines how the training sample is split at each node. There are five settings available, but we find the default Gini-Index to produce the best results. Finally, for regression tasks, the `PruneMethod` can either be cost complexity or nothing. We use the cost complexity method to recover from overtraining.

With selected variables and optimized parameters for MLP, BDT (AdaBoost.R2), and BDT (GradientBoost) MVA methods, we feed the $\sim 7 \times 10^5$ PYTHIA MC generated events into TMVA. Half of these events are used for training and half for testing.

RESULTS

For each MVA method, we build a complicated nonlinear function from the set of selected variables into the set of possible m_h values through training. Then, that function's

performance is evaluated through *testing*, in which the remaining half of the events have their variables mapped to reconstructed Higgs masses ($m_{h,reco}$). From TMVA we ultimately receive $\sim 3 \times 10^5$ pairs of $m_{h,reco}$ and $m_{h,true}$ data. These results can be depicted in three ways.

First, the total MVA method RMS can be calculated by RMS $= \sqrt{\langle {d_{reg}}^2 \rangle}$. This RMS can then be compared for multiple methods to determine which method has the overall least difference between reconstructed and true Higgs mass. The RMS values for the MLP, the BDT with AdaBoost.R2, and the BDT with GradientBoost methods are presented in Table 2 along with some extra information. Note that half of the values in Table 2 are "truncated," meaning they were calculated from only those events whose d_{reg} values lie within two standard deviations of the mean, i.e., in the middle 95%. On the other hand, the truncated RMS roughly represents the difference between $m_{h,reco}$ peak values and $m_{h,true}$ values. Finally, while $\langle d_{reg} \rangle$ is sensitive to sign, comparing the truncated and non-truncated entries for a given method provides shape and location information about the d_{reg} plot. For instance, the magnitude and sign of the difference indicates the degree and sign of the skewness of d_{reg}. Additionally, the size of $\langle d_{reg} \rangle_T$ reflects how centered about $m_{h,true}$ the $m_{h,reco}$ plot is.

MVA Method	*RMS* (MeV)	$\langle d_{reg} \rangle$ (MeV)	RMS_T (MeV)	$\langle d_{reg} \rangle_T$ (MeV)
MLP	1.25×10^4	47	9.73×10^3	1.21×10^3
BDT with AdaBoost.R2	1.38×10^4	631	1.25×10^4	1.29×10^3
BDT with GradientBoost	1.24×10^4	-1.49×10^3	8.99×10^3	-281

Table 2: A performance summary for all MVA methods utilized.

Second, we can plot d_{reg} on its own and against $m_{h,true}$ (Figure 7). These plots tell us whether the MVA method is under- or over-estimating the Higgs mass and for which $m_{h,true}$. Third and finally, for any one of 90, 100, 110, 120, 130, 140, or 150 GeV, we can plot both the reconstructed Higgs mass (Figure 8).

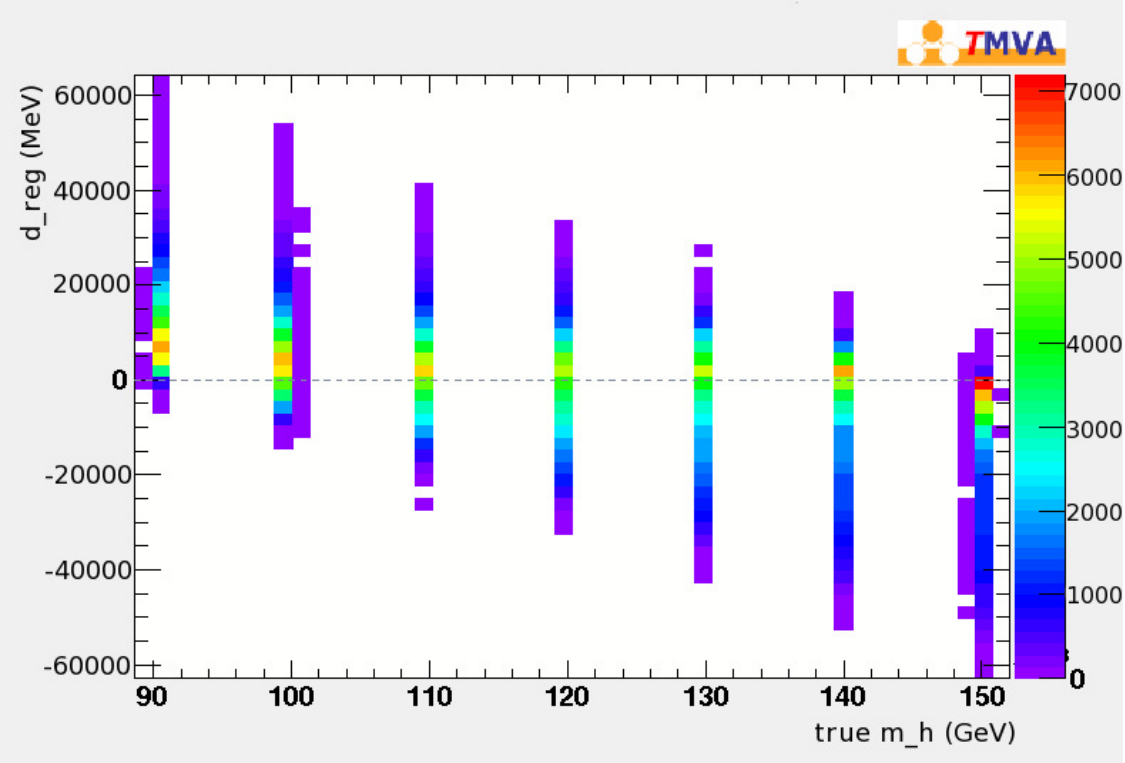


Figure 7: Plot of d_{reg} vs. $m_{h,true}$ for a BDT with GradientBoost.

DISCUSSION AND CONCLUSIONS

In order to visualize the actual performance of our trained MVA methods, we compare the reconstructed Higgs mass plot (Figure 8) to the PYTHIA MC generated $m_{h,true}$ plot, the $m_{h,reco}$ resulting from a single mass-reconstruction algorithm, and the current best NN-reconstructed Higgs mass plot all for events generated with an inputted $m_{h,true}$ of 120 GeV (Figure 9). We take this state-of-the-art in m_h reconstruction to be Barklow's [2], which uses the `TMultiLayerPerceptron` class in ROOT with multiple algorithmically reconstructed Higgs masses as inputs. This allows us to witness the development of m_h reconstruction methods from single algorithms to coarse, ROOT-based neural nets to powerful BDTs.

First, we notice that the PYTHIA generated $m_{h,true}$ plot has a width of 12.3 MeV, reflecting the narrow intrinsic Higgs mass width. Second, the m_h reconstruction using a single

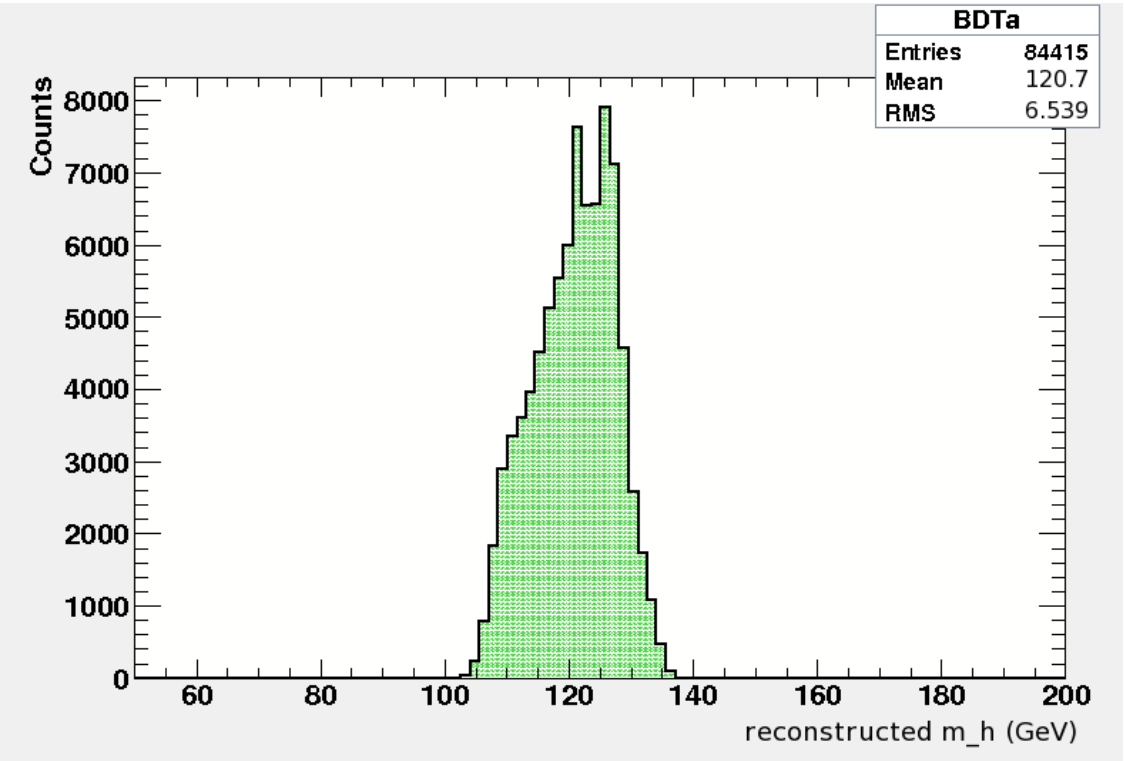
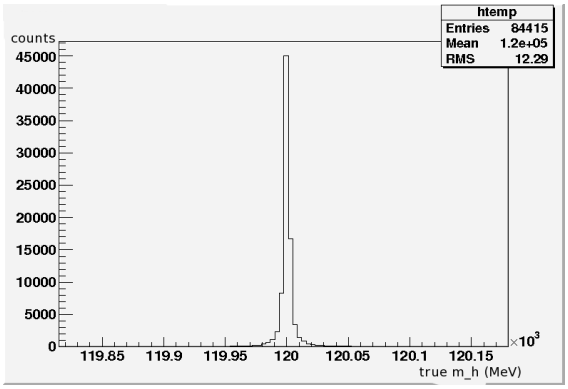


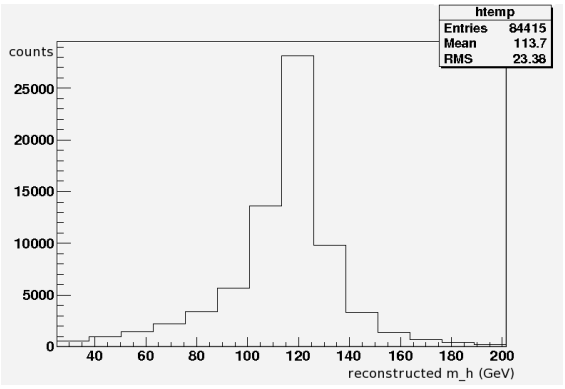
Figure 8: Plot of $m_{h,reco}$ for a BDT with AdaBoost.R2 using PYTHIA MC generated signal events with a 120 GeV SM Higgs boson.

algorithm shows a peak at $\sim$120 GeV, but a mean of 113.7 GeV and a width of 23.4 GeV. Third, the NN m_h reconstruction plot has its peak and mean at 120 GeV but a width of 8.6 GeV. Finally, our m_h reconstruction using a BDT with AdaBoost.R2 peaks around 120 GeV, has a mean of 120.7 GeV, and has a width of 6.5 GeV. We can therefore say that our MVA methods yield competitive m_h resolutions when compared to all previous approaches.

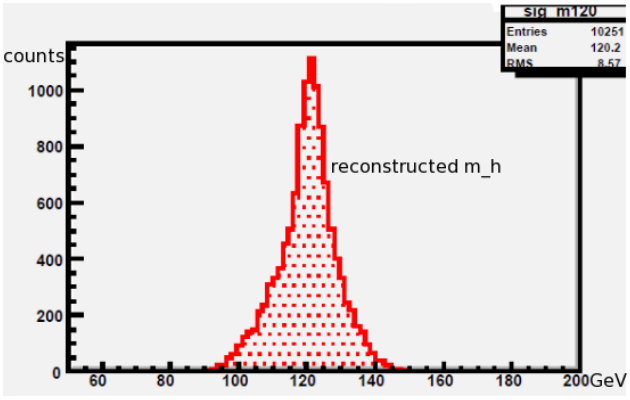
Although the $b\bar{b}$ Higgs decay signal was by no means extracted and observed, this reconstruction attempt was a success. Not only were gains made in Higgs mass resolution, but they were made as a proof-of-principle. Many elements of the development of the m_h-reconstruction method were crude and abbreviated — there is much room to improve our method. We have merely shown that it may be possible to construct a very effective m_h-reconstruction algorithm using MVA methods.



(a) PYTHIA generated $m_{h,true}$ plot with an inputted Higgs mass of 120 GeV.



(b) Single algorithm reconstructed m_h from PYTHIA generated signal events with a 120 GeV Higgs boson.



(c) Previous state-of-the-art NN-reconstructed m_h from signal events with a 120 GeV SM Higgs boson generated by ALPGEN (an MC particle physics generator best for low-p_T events).

Some of the improvements that can, and should, be made are in event generation, event processing, and the training and testing of MVA methods. Immediately, we should generate $gg \to h \to b\bar{b}$ events with more than seven distinct values for $m_{h,true}$ to reduce discretization in the final $m_{b\bar{b}}$ plots. As more results from the LHC are released, we could also generate events for additional Higgs decay modes as well as with beyond-the-SM physics. Another relatively easy improvement would be to optimize the MVA-method-specific parameters as well as select variables more gracefully. In both cases, the methods we use are crude and insensitive — algorithms may be used, for example, for both of these purposes. Additionally,

Hoecker et al. [8] are continually working to improve TMVA and include more advanced MVA methods as well as combined MVA methods; these should be trained and tested as soon as they are released.

Finally, the largest potential improvement will be made from upgrading our event processing. Specifically, we will initially process events with a broader method of defining variables, allowing them to be selected from a pool of $\sim$700. With this added information, the MVA methods may be able to find even subtler patterns in the events, increasing their ability to resolve the true Higgs mass. In conclusion, we are encouraged by our MLP and BDT results — they show that MVA methods can be very useful for m_h discrimination in ATLAS.

ACKNOWLEDGMENTS

This work was carried out at SLAC National Accelerator Laboratory from June–August 2011 as part of the Summer Undergraduate Laboratory Internship (SULI) program created, organized, and funded by the U.S. Department of Energy, Office of Science. I would like to thank my mentor Tim Barklow as well as Steve Rock, the director of the SULI program. Lastly, I would like to thank the ATLAS collaboration at the LHC for access to their resources.

REFERENCES

[1] M. Takahashi, "Standard Model Higgs Searches at the LHC," 2008, arXiv:hep-ex/0809.3224.

[2] T. Barklow, "Higgs $\rightarrow$ bb." `http://indico.cern.ch/getFile.py/access?contribId=40&sessionId=5&resId=0&materialId=slides&confId=116471`, Jan. 2011.

[3] B. Martin and G. Shaw, *Particle Physics*. Manchester Physics Series, Wiley, 2008.

[4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, New York: Springer, 2001.

[5] F. Hakl, M. Hlaváček, and R. Kalous, "Application of neural networks to Higgs boson search," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 502, no. 2-3, pp. 489 – 491, 2003.

[6] J. Bastos, "Tagging heavy flavours with boosted decision trees," 2007, arXiv:physics.data-an/0702041.

[7] T. Sjöstrand, S. Mrenna, and P. Skands, "PYTHIA 6.4 Physics and Manual," *JHEP*, vol. 05, p. 026, 2006.

[8] A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, and H. Voss, "TMVA: Toolkit for Multivariate Data Analysis," *PoS*, vol. ACAT, p. 040, 2007.

[9] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*. The MIT Press, 1995.