SLAC-TN-06-015

Using Boosted Decision Trees to Separate Signal and Background in $B \to X_s\gamma$ Decays

James Barber

Office of Science, SULI Program

University of Massachusetts, Amherst

Stanford Linear Accelerator Center

Stanford, California

August 23, 2006

Prepared in partial fulfillment of the requirements of the Office of Science, U.S. Department of Energy Science Undergraduate Laboratory Internship (SULI) Program under the direction of Philip Bechtle at the Stanford Linear Accelerator Center.

Participant: _____

Signature

Research Advisor: _____

Signature

Work supported in part by Department of Energy contract DE-AC02-76SF00515

TABLE OF CONTENTS

ABSTRACT

Using Boosted Decision Trees to Separate Signal and Background in $B \to X_s\gamma$ Decays. JAMES BARBER (University of Massachusetts, Amherst, Amherst, MA 01003) PHILIP BECHTLE (Stanford Linear Accelerator Center, Stanford, CA 94025)

The measurement of the branching fraction of the flavor changing neutral current $B \to X_s\gamma$ transition can be used to expose physics outside the Standard Model. In order to make a precise measurement of this inclusive branching fraction, it is necessary to be able to effectively separate signal and background in the data. In order to achieve better separation, an algorithm based on Boosted Decision Trees (BDTs) is implemented. Using Monte Carlo simulated events, 'forests' of trees were trained and tested with different sets of parameters. This parameter space was studied with the goal of maximizing the figure of merit, Q, the measure of separation quality used in this analysis. It is found that the use of 1000 trees, with 100 values tested for each variable at each node, and 50 events required for a node to continue separating give the highest figure of merit, Q = 18.37.

INTRODUCTION

A rare B meson decay explored at BaBar is the process $B \to X_s\gamma$, in which a B meson decays to a photon and a hadronic final state containing an s quark from the $b \to s\gamma$ transition. A Feynman diagram of this transition is given in Figure 1. This process is predicted to have a branching fraction of $\mathcal{B}(B \to X_s\gamma) = (3.61 \pm 0.49) \times 10^{-4}$ [1], which agrees well with the world average of previous measurements of $\mathcal{B}(B \to X_s\gamma) = (3.55 \pm 0.26) \times 10^{-4}$ [2]. With a future improvement in the theoretical uncertainty expected, an improvement in the precision of the experimental measurement would increase the sensitivity for new physics.

The $B \to X_s\gamma$ measurement presented here is fully inclusive, meaning any final state is allowed. This has the advantage of reduced theoretical uncertainties compared to exclusive measurements. These uncertainties stem from the predictions of specific final states, such as $B \to K^*\gamma$, and are due to uncertainties in the fragmentation, i.e. the calculation of how the remaining quarks combine to form hadrons. These uncertainties are largely avoided by allowing for all possible hadronic states. Reduced information about the kinematics of the entire final state, however, makes background suppression difficult because neither of the two B mesons in the event are reconstructed.

The background is split into two categories: $B\bar{B}$ and continuum. $B\bar{B}$ background refers to decays from $B\bar{B}$ events not of the type $B \to X_s\gamma$. Continuum background, by contrast, is comprised of all non-$B\bar{B}$ events and is present at and below energies of $B\bar{B}$ events. To gain information about the continuum background, 10% of the data is taken at energies too low for $B\bar{B}$ production. Thus all data taken at 'off-peak' energies is continuum, while 'on-peak' data is a mix of continuum and $B\bar{B}$ backgrounds. Expected amounts of background and signal data are shown in Figure 2a, where it can be seen that the amount of background must be reduced by 3 orders of magnitude in order to detect a significant signal. Figure 2b shows the result of a Fisher discriminant method of separation using event-shape variables and lepton tagging to separate signal and background. Lepton tagging is a method of identifying

$B\bar{B}$ events by requiring the presence of a high momentum lepton (from the decay $B \to Xl\nu$) in order to accept an event. The higher the momentum of the lepton is required to be, the less likely it is to have come from a continuum event. The branching fraction of a semi-leptonic B decay is, however, only about 10%, so in using this method at least 90% of $B\bar{B}$ events are rejected. The Fisher discriminant method of signal and background separation, used after lepton tagging, successfully eliminated 99% of the background but also rejected 95% of the signal. To reduce the amount of signal rejected, more advanced techniques for the suppression of both the continuum and $B\bar{B}$ backgrounds are needed. With these advanced techniques it is possible to use both the event-shape and lepton tagging variables concurrently to increase the selection efficiency.

The current technique used in the extraction of signal from background data is based on an Artificial Neural Network (ANN). When variables with low individual separation power and non-negligible, non-linear correlations (such as the event-shape variables used in this analysis) are used to separate signal and background, ANNs outperform methods based on a likelihood estimator or Fisher discriminant. In order to obtain high separation from variables with low separation power, an ANN must be 'trained' on Monte Carlo (MC) simulated data. By varying the importance of different variables (via adjusting the variables' weights) and combining variables in different manners, an ANN can learn which of these variable combinations and weight values yield the highest ratio of signal to background. The variable combination and weight adjustment is done in hidden layers of the ANN (as shown in Figure 3), the exact process of which is completely concealed from the user. This learning process, of re-weighting and re-combining variables in order to achieve the best signal and background separation, is referred to as 'training'.

One of the drawbacks in using an ANN is that it is possible to decrease the total separation power by giving the ANN too many variables with low separation power. Another drawback of an ANN is that ANNs are very sensitive to their training; the order of input variables in the training may affect the output, and if the Monte Carlo sample used to train the ANN

does not very closely model the real data, the separation power of an ANN can suffer more than that of other methods.

In order to avoid these problems, an algorithm implementing Boosted Decision Trees (BDTs) has been developed to perform the signal and background separation. This method, like an ANN, must be trained to yield the best separation of signal and background (see [3] for a more detailed explanation of this process). To train a BDT, a number of MC events are chosen as training events and put into a 'root node'. The algorithm then iterates through each variable, finding the value at which a selection would give the highest signal and background separation. The variable that would give the highest separation is chosen and the training events are subjected to this selection. Events are sent to the left or right child, depending on whether the selection classifies them as signal or background. At each new node a variable and value are chosen from which a selection is made. The events are again separated and the process is repeated. Nodes are separated until they contain less than some minimum number of events or have a signal to background (or background to signal) ratio greater than a given limit. Nodes are then classified as either 'signal' or 'background' depending on whether the majority of events in that node is signal or background, respectively (see Figure 4). Misclassified signal and background events (e.g. signal events in background nodes or vice versa) are given an increased weight and the entire process is started again, with a new root node established and a new tree created. The result of increasing the weights of these previously misclassified events is that these events become more important when determining signal to background separation in the next iteration. In this way, a specified number of trees is created and the training is complete.

After training, the resulting 'forest' of trees must be tested to determine how well it separates signal and background. Events chosen for testing (of which none were used for training) are sent through each tree in the forest, and for each event a likelihood value is calculated. This value is equal to the number of times an event ends in a signal node divided by the number of trees it is sent through. An event classified as signal by every tree would

thus have a likelihood value of 1, whereas an event always classified as background would have a likelihood value of 0. In this way, signal events tend to bunch closer to 1 than 0, while background events tend toward 0. A threshold value is determined, which is equal to the likelihood value about which a selection made would give the best signal and background separation.

In order to determine superiority of one method over another, we must have some way to compare the quality of separation. We calculate a figure of merit, Q, for each separation method:

$$Q = \frac{S}{\sqrt{S+B}},$$

where S is the number of signal events correctly classified as signal and B is the number of background events incorrectly classified as signal. Because we are dealing with two types of background – $B\bar{B}$ and continuum – B is defined as

$$B = N_B(1 + f \cdot B_{MCerror}) + \frac{N_C}{1-f},$$

where N_B is the number of $B\bar{B}$ background events and N_C is the number of continuum background events. The factor f is the on-peak fraction, i.e. the fraction of data taken at the energy required for $B\bar{B}$ pair production. In typical BaBar data taking, f is about 90%. Hence, (1 - f) is the fraction of available off-peak data from which continuum measurements are made. The term $B_{MCerror}$ accounts for systematic uncertainties in the MC generated $B\bar{B}$ background events.

Once the forest of trees with the highest figure of merit has been formed, it is ready to be used with real data. Each data event is sent through the forest of trees and, just like with testing events, a likelihood value is determined. If the likelihood value is above the previously determined threshold value, the event is classified as signal.

MATERIALS AND METHODS

Rather than creating a new implementation of the BDT algorithm, modifications were made to an existing implementation contained in the Toolkit for Multivariate Analysis (TMVA) [4]. A program was created to interface with the TMVA package, allowing us to input data. As mentioned above, BDTs must be 'trained' to separate signal and background and then tested to measure their efficiencies. Because of this, parameters indicating the number of events to use for training need to be passed into the program along with the data used in the training and testing. It is important for the BDT to be trained on a sample of data representative of the entire data range so that variations in the data will be accounted for by the BDT. It is also important that the testing sample be statistically independent of the training sample and taken from the same data range to ensure proper measurement of separation efficiency. In this way, the agreement of the results of the training and testing can be used to ensure that the BDT does not classify the data according to particular features of the training sample (known as "over-training") but separates based on general event properties of the signal and background.

It was found that the method used by the TMVA to select training and testing events was not appropriate for our set of input events, because the events chosen were not representative of the entire event range. Let us take, for instance, a sample of 100,000 (N_{total}) events, with 10,000 (N_{train}) used for training and 50,000 (N_{test}) used for testing. The TMVA would use events numbered 1-10,000 for training and events numbered 10,001-60,000 for testing. If different experimental setups were used to collect data, and different MC data samples were generated to reflect those changes, testing could be done on a continuous subset of events representative of only a portion of the actual data. In this case, the calculated quality of separation would be inaccurate. It is then obvious that training and testing event samples must be selected from throughout the entire data set to protect against training and testing on events from different experimental setups. To fix this problem, changes were made to the

TMVA code to ensure that events used for training and testing are taken from the entire data range. To do this, we required that if N_{train} events were asked for, approximately one out of every $\frac{N_{total}}{N_{train}}$ events is selected for training.

A second issue in the selection of events that needed to be addressed concerned the ratio of signal to background events. If the ratio of signal to background events in our Monte Carlo data sample is $\frac{N_{sig}}{N_{bkg}}$, it is imperative that our training sample has this same ratio of signal to background. As implemented, the TMVA used equal numbers of signal and background events. Failure to retain the proper signal to background ratio would cause incorrect values of the figure of merit to be calculated, thus giving an incorrect assessment of the separation power of the algorithm. The algorithm used to select events was again changed to accommodate the needs of our analysis.

With the aforementioned problems fixed, we began training and testing forests of trees using different parameter values in order to find the setup yielding the maximal figure of merit. From a data set of 749,684 MC events, 100,000 events were chosen for training. The number of trees in the forest (N_{trees}) was either 500 or 1,000; the minimum number of events ($N_{minEvents}$) required for a node to be separated further was either 50 or 100; and the number of values checked for each variable in determining how to best separate a node (N_{cuts}) was either 25, 50, or 100. At the conclusion of testing, figures of merit were calculated for each parameter setup so that the separation quality of each might be compared.

RESULTS

Table 1 shows the different parameter setups for which trees were trained and tested, as well as the resultant figure of merit for each setup. As can be seen, parameter set 'b' is found to give the highest figure of merit. The results of the testing conducted with this parameter set are shown graphically in Figure 5, where the likelihood value is plotted on the x-axis in red for signal, blue for $B\bar{B}$ background, and cyan for continuum background. The vertical line

superimposed on the graph shows the threshold value at which a selection would yield the maximal figure of merit. This threshold value was found to be .595, which gave a figure of merit, Q, of 18.37. Plots representative of the selection quality are shown in Figures 6 and 7. Figure 6 shows the total number of signal and background events both before and after the selection algorithm. Figure 7 shows the efficiency of selection for the $B \rightarrow X_s\gamma$ signal as well as both the $B\bar{B}$ and continuum backgrounds. Note that a low efficiency is desirable for the backgrounds, as the efficiency is a measure of the number of events selected as signal.

DISCUSSION AND CONCLUSIONS

Table 1 suggests that the figure of merit increases as the number of trees (N_{trees}) is increased, and also as the number of values (for each variable) used to determine where to best make a selection (N_{cuts}) is increased. The results also indicate an increase in the figure of merit as the minimum number of events required in a node ($N_{minEvents}$) is decreased, however this effect is small compared to the effect due to an increase in either N_{trees} or N_{cuts}. The gain resulting from an increase in these two variables, however, comes at the cost of speed. The number of trees selected should scale linearly (as a roughly constant amount of computing must be done to train a tree), however increasing the value of N_{cuts} greatly increases the time it takes to train each tree. This increase in computational time for each tree is so great when compounded over 1000 trees that it took 5 times longer to run the program with parameter setup 'b' than with setup 'a'.

To determine whether this BDT algorithm is preferable to the ANN currently used by the analysis requires the generation of comparable figures of merit for each method. This can be realized by designing within a Monte Carlo event sample a group of events designated for training and another set of events designated for testing. In running the BDT and ANN on this same set of events and calculating figures of merit for each method, a direct comparison can be made.

It is hoped that the result of such a comparison will show the BDT implementation worked on this summer superior to the ANN currently in use, and that the use of BDTs in separating signal and background will lead to increased precision in the analysis of the $B \to X_s\gamma$ decay. This precision will help the search for physics beyond the standard model as it puts tighter limits on the experimental value for this branching fraction. If it is found that the theoretical predictions and experimental measurements for this branching fraction do not agree it may point to the presence of an until now undetected massive particle, such as a Higgs or SUSY particle, in the radiative penguin loop.

ACKNOWLEDGMENTS

I would sincerely like to thank my mentor Philip Bechtle for his guidance throughout this project and his eagerness to discuss all aspects of particle physics, not merely those pertaining to my research. I would also like to thank Rainer Bartoldus for his thoughts and suggestions on how to best present my work. I would like to acknowledge the U.S. Department of Energy, Office of Science and all those at the Stanford Linear Accelerator Center for giving me the opportunity to participate in the SULI program. Thanks to everyone involved, it has truly been a rewarding and educational experience.

REFERENCES

[1] Heavy Flavor Averaging Group (HFAG) and E. Barberio, *et al.*, "Averages of b-hadron properties at the end of 2005," 2006. http://www.slac.stanford.edu/xorg/hfag/

[2] T. Hurth, E. Lunghi and W. Porod, "Untagged $B \to X_{sd}\gamma$ CP asymmetry as a probe for new physics," *Nucl. Phys.*, vol. B704, pp. 56-74, 2005.

[3] B. P. Roe, *et al.*, "Boosted decision trees, an alternative to artificial neural networks," *Nucl. Instrum. Meth.* vol. A543, pp. 577-584, 2005.

[4] TMVA package developed by Andreas Höcker (CERN), Jörg Stelzer (CERN), Helge Voss (MPI-KP Heidelberg), Kai Voss (U. of Victoria), and Xavier Prudent (LAPP-Annecy). http://tmva.sourceforge.net/

FIGURES

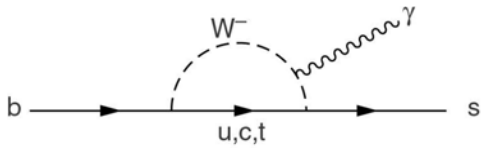
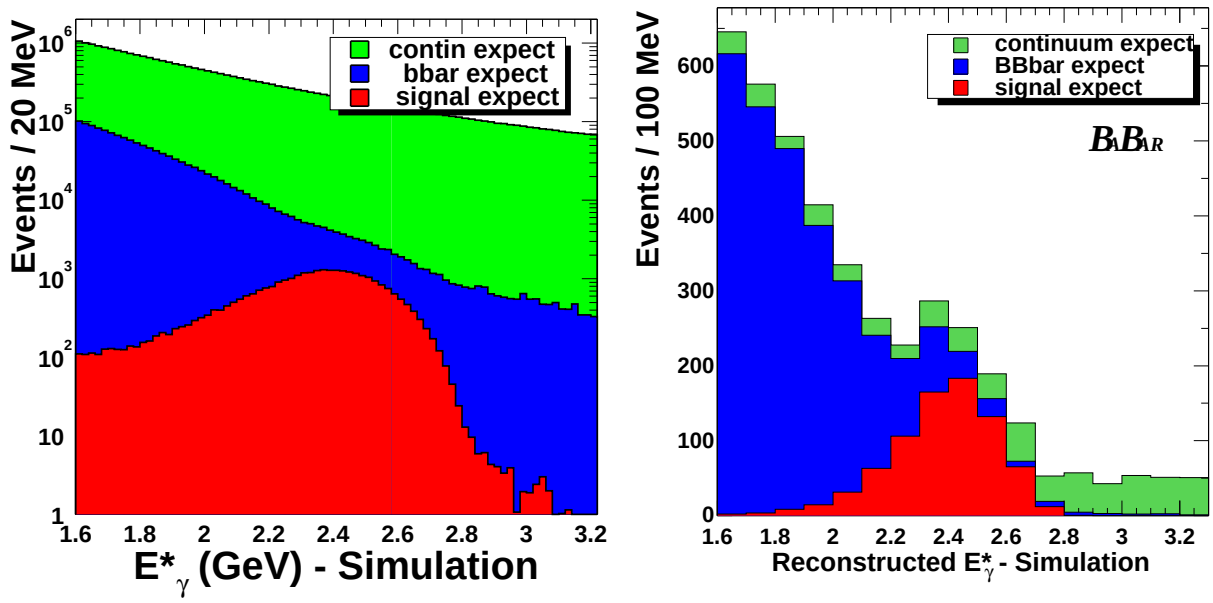


Figure 1: Feynman diagram of $b \rightarrow s\gamma$ transition.

Parameter setup a		**Parameter setup b**	
N_{trees}	1000	N_{trees}	1000
$N_{minEvents}$	50	$N_{minEvents}$	50
N_{cuts}	50	N_{cuts}	100
figure of merit:	18.21	figure of merit:	18.37
Parameter setup c		**Parameter setup d**	
N_{trees}	500	N_{trees}	1000
$N_{minEvents}$	50	$N_{minEvents}$	100
N_{cuts}	50	N_{cuts}	50
figure of merit:	18.04	figure of merit:	18.18
Parameter setup e			
N_{trees}	500		
$N_{minEvents}$	50		
N_{cuts}	25		
figure of merit:	17.91		

Table 1: Parameter setups and corresponding figures of merit



(a) (b)

Figure 2: Signal and background plotted before (a) and after (b) selection including the Fisher method of separation.

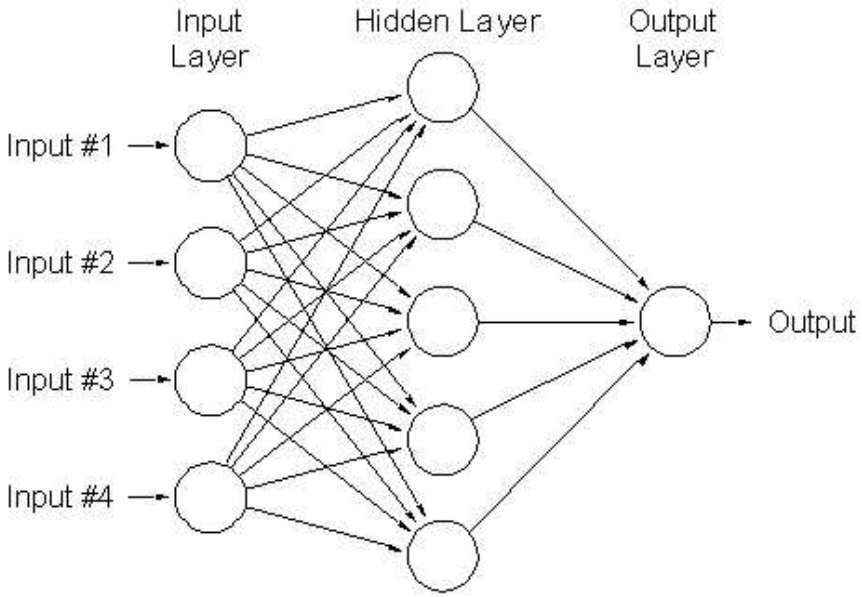


Figure 3: Variable weighting and combination are done in the hidden layers of an ANN

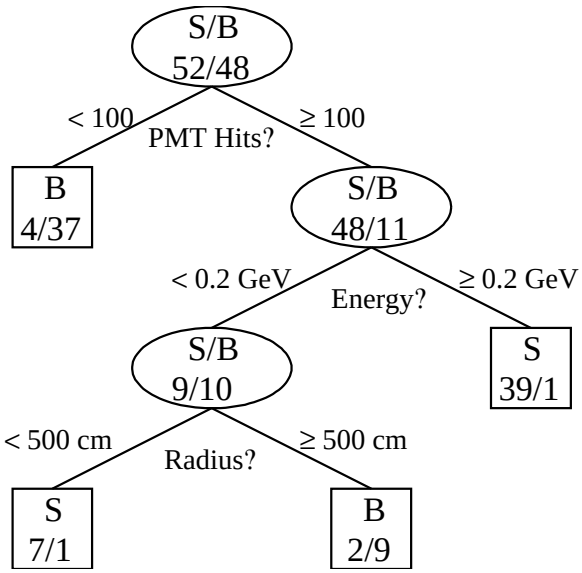


Figure 4: Example splitting of a BDT

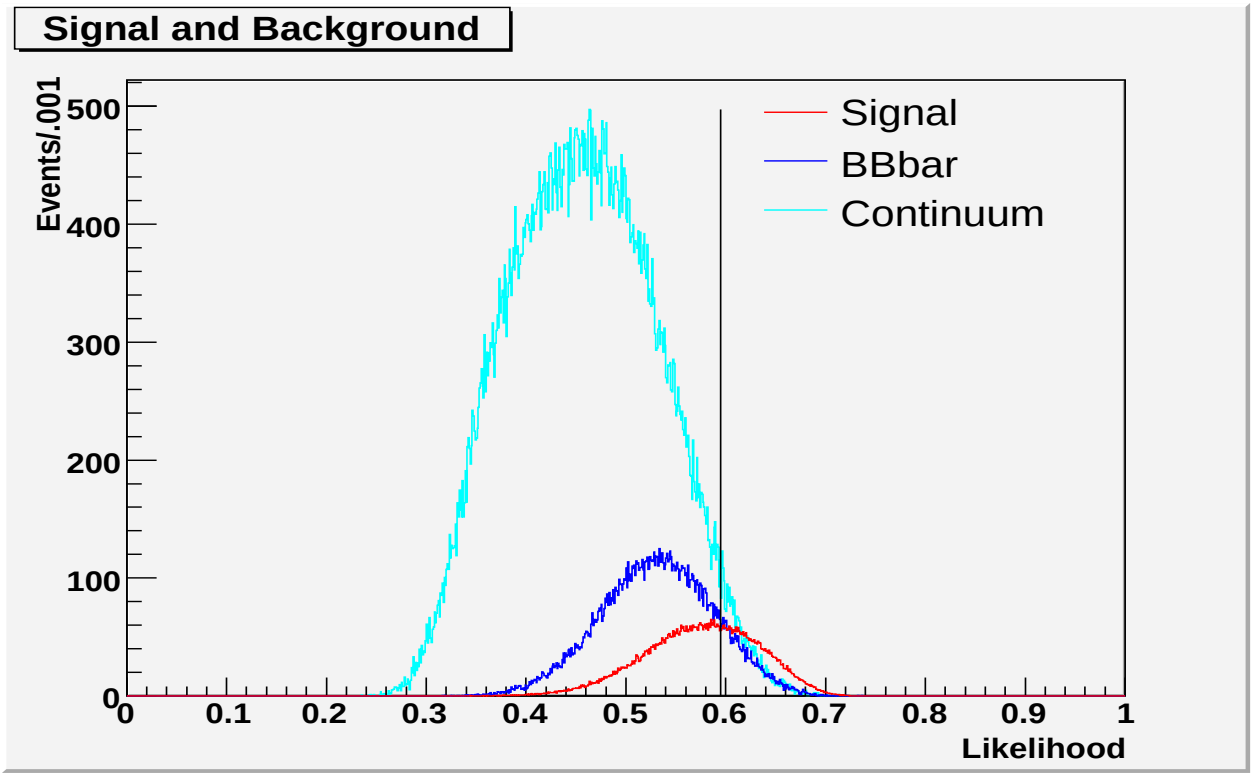
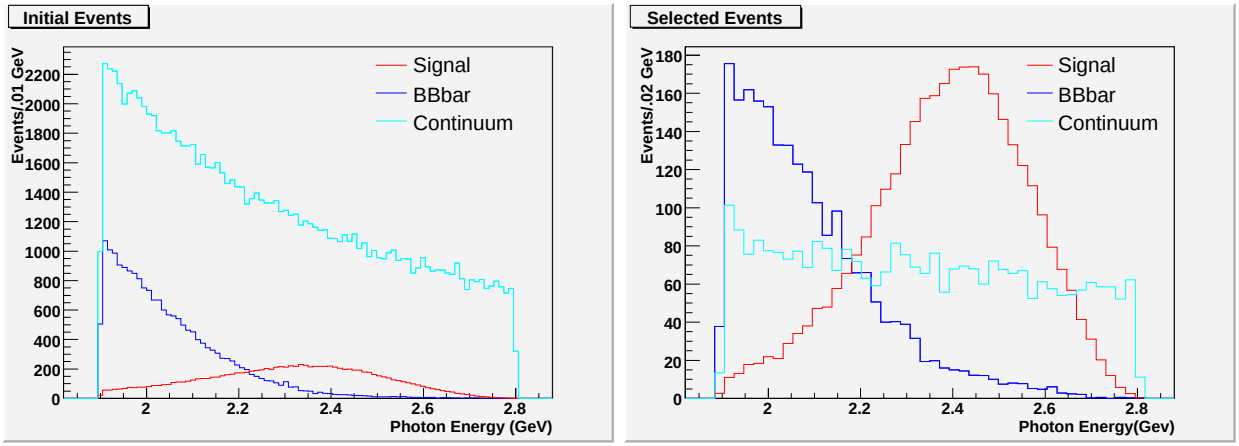


Figure 5: Likelihood values for Monte Carlo testing events, using parameter set 'b'. A selection at .595 (indicated by verticle line) gives a figure of merit of 18.37.



(a) (b)

Figure 6: Number of signal and background events before (a) and after (b) BDT separation with parameter set 'b'.

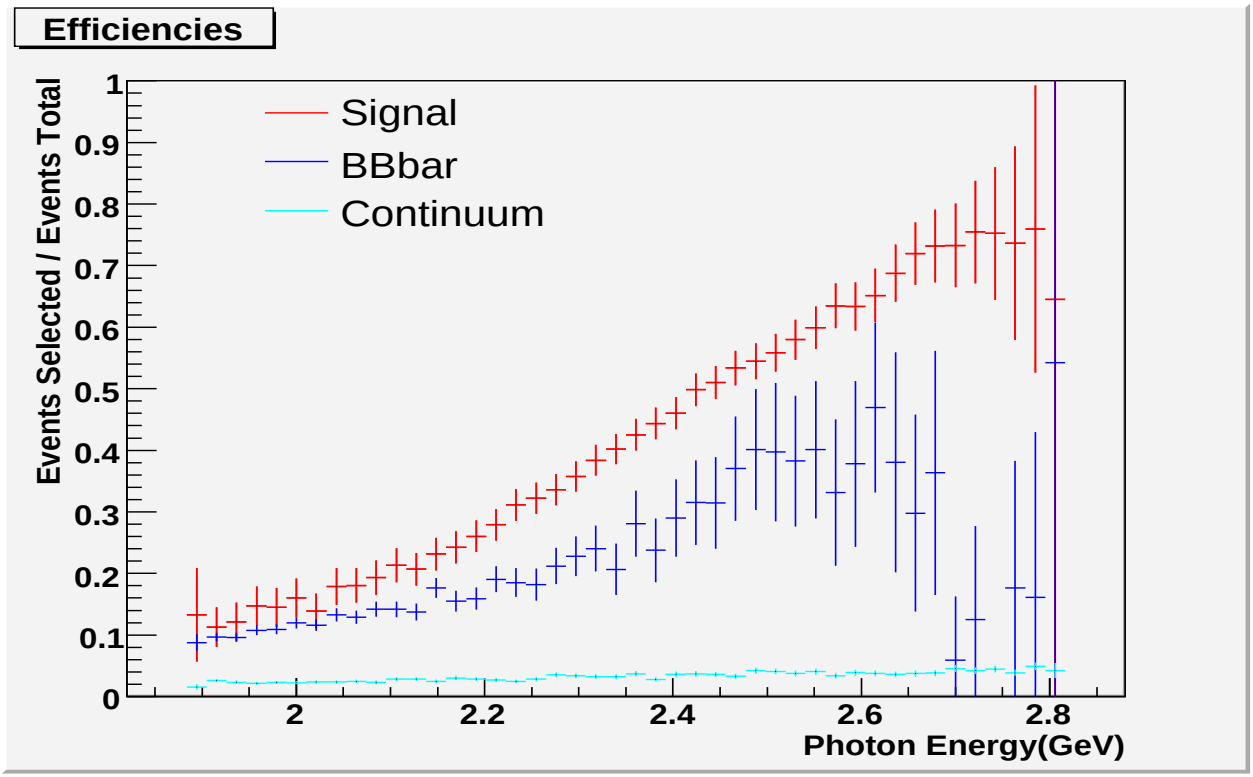


Figure 7: Signal and background efficiencies of BDT algorithm with parameter set 'b'.