Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The $B^0 \to K^+\pi^-e^+e^-$ selection in the LHCb experiment

Master's Thesis

Ya Zhao

Departement of Physics, D-PHYS
ETH Zürich

Supervisors:
Prof. Dr. Nicola Serra(UZH)
Dr. Rafael Silva Coutinho(UZH)
Dr. Patrick Owen(UZH)

September 16, 2021

Abstract

This analysis studies the selection of $B^0 \to K^+\pi^-e^+e^-$ in the LHCb experiment. The backgrounds from $B^0 \to K^*(\to K^+\pi^-)J/\psi(\to e^+e^-)$ and $\Lambda_b^0 \to pK^-J/\psi(\to e^+e^-)$ are discussed. The cut-based selection method is used firstly to veto the background and the best cuts are found. Then selections using a BDT method are studied. Finally the results from these two methods are compared and the differences are shown.

Contents

Chapter 1

Introduction

1.1 LHCb experiment

The LHCb experiment is situated at one of the four points around CERN's Large Hadron Collider, which accelerates two proton beams to a center-of-mass energy of 13 TeV. The LHCb detector is a single-arm forward spectrometer covering the pseudorapidity range $2 < \eta < 5$. The aim of the LHCb experiment is to record the decay of particles containing b and anti-b quarks and the primary purpose of LHCb is to search for new physics in CP violation and rare decays of beauty and charm hadrons, and this requires LHCb has excellent tracking performances in terms of momentum, primary vertex resolution, excellent particle identification capabilities and so on[1].

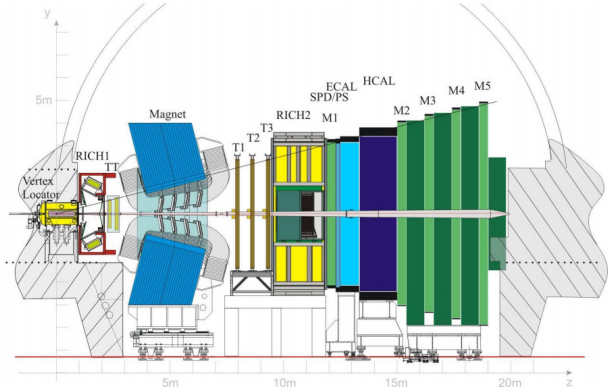

Figure 1.1: LHCb detector[1]

The tracking system[2] consists of the vertex locator(VELO) and four planar tracking stations. The VELO is situated around the interaction region inside a vacuum tank, and the four stations are the Tracker Turicensis(TT) upstream of the dipole magnet and T1-T3 downstream of the magnet[3]. The VELO contains 42 silicon modules arranged along the beam, each providing a measurement of the r and ϕ coordinates. The TT and IT detectors use silicon microstrip sensors with a strip pitch of $183\mu m$ and $198\mu m$, respectively. The Outer Tracker is a drift-tube gas detector consisting of approximately 200 gas-tight straw-tube modules with drift-time read-out. Charged hadron identification in the momentum range from 2 to 100GeV/c is achieved by two Ring Imaging Cherenkov detectors(RICH1 and RICH2) read out by Hybrid Photon Detectors(HPDs). The calorimeter system is composed of a Scintillating Pad Detector(SPD), a Preshower(PS), a shashlik type electromagnetic calorimeter(ECAL) and a hadronic calorimeter(HCAL). It provides the identification of electrons, photons and hadrons as well as the measurement of their energies and positions, and selects candidates with high transverse energy for the first trigger level(L0). The muon detection system provides muon identification and contributes to the L0 trigger of the experiment.

The LHCb data-flow consists of online and offline processes. The PP collisions rate is much higher than the readout frequency of LHCb tracking detectors, so several trigger steps are needed to make sure that the information could be stored permanently. The LHCb trigger system consists of two levels: the first level is implemented in hardware and is designed to reduce the event rate from the nominal LHC bunch crossing rate of 40 MHz to a maximum of 1.1 MHz, and the second level is software trigger, High Level Trigger(HLT). Then the offline step processes the online-selected data to achieve the best reconstruction quality for the analysis of data. After reconstruction[4], events are further selected and categorized by a stripping procedure through stripping line which corresponds to a single process of interest.

1.2 Flavor changing neutral current processes in B decays

The flavor changing neutral current processes(FCNC) include particle-antiparticle mixing, certain rare and radiative meson decays, CP-violating decays and so on. The FCNC processes have played an important role in the Standard Model, and these processes are governed by the GIM mechanism[5] which can suppress these processes naturally. As a result, there are no FCNC processes at the tree level, and the one-

loop diagrams, the penguin and box diagrams, have the leading contributions. This fact makes the FCNC processes a very powerful tool for the determination of some parameters of the CKM matrix[6][7], for example the top-quark couplings $|V_{td}|$ and $|V_{ts}|$ and the CP-violating phases. Besides, the FCNC processes are sensitive to the contribution from physics beyond the Standard Model, thus they are also a good tool to study new physics.

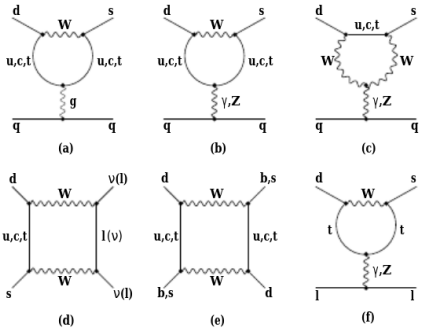

Figure 1.2: Some typical Penguin and Box diagrams[8].

At present only a few FCNC transitions have been observed experimentally[8]:

- $K^0 - \bar{K}^0$ mixing, the related $K_L - K_S$ mass difference and the indirect CP violation in $K_L \to \pi\pi$ represented by the parameter ϵ_K.
- $B_d^0 - \bar{B}_d^0$ mixing and the related mass difference $(\Delta M)_d$
- $K_L \to \mu\bar{\mu}$
- $B \to X_s\gamma$ and $B \to K^*\gamma$.

The interest in FCNC decays can be formulated as follows[9]:

- They measure the parameters of the CKM matrix and play a central role in testing the unitarity of this matrix.
- They are vitally important in measuring CP violation in flavor changing processes.
- They offer fertile testing grounds for Quantum Chromodynamics(QCD); the available techniques(Perturbative QCD, Heavy quark effective theory HQET, Lattice-QCD, QCD-sumrules) are directly applicable in these decays.

- They may reveal new physics, such as supersymmetry SUSY, more generations, leptoquarks, etc.

- Most importantly, they are accessible at the present and planned experimental facilities.

The transition of a bottom quark into a strange quark is an important example of the FCNC processes, which can occur through the same quantum loop transition of the GIM mechanism but is dominated by the contribution from the top quark[10]. A particular rare b-hadron FCNC decay involves the decay of a B_s^0 hadron into a pair of muons. Another important FCNC process involves the decay of a bottom quark into a strange quark and two leptons, $b \to sll$. These processes are called semileptonic decay, because the decay products include leptons and hadrons. Measurement of the properties of these decays are sensitive to new particles with masses up to around 100 TeV[11].

For these decays, the simplest property to measure is the branching fraction[10]. For semileptonic decays, the branching fraction is measured as a function of the squared four-momentum transferred to the two leptons. The influence of new physics depends on the squared four-momentum, thus this is an important variable. Besides, the ratio of branching fraction between decays involving electrons and muons can be considered. The mass of the muon is about 200 times heavier than the mass of the electron, but except that the muon has exactly the same properties as electron in the Standard Model. This phenomenon is known as the Lepton Universality. Looking at the ratio of the branching fraction is a sensitive way to search for new physics, since new particles which are not predicted by the Standard Model don't have to follow the Lepton Universality. In rare b-hadron decays, one famous example is the ratio $R_K = B(B^+ \to K^+\mu^+\mu^-)/B(B^+ \to K^+e^+e^-)$. This ratio is known as to be unity in the Standard Model with very high precision[12][13][14]. Any measurement that deviates significantly from unity would be an unambiguous sign of physics beyond the Standard Model. In addition to studying b-hadron decays into a specific final state, one can also consider the inclusive $b \to sll$ process, where the final state can comprise any number of hadrons.

Another important ratio $R_{K^{*0}}$ is defined by:

$$R_{K^{*0}} = \frac{\int \frac{d\Gamma(B\to K^{*0}\mu^+\mu^-)}{dq^2} dq^2}{\int \frac{d\Gamma(B\to K^{*0}e^+e^-)}{dq^2} dq^2}.$$

A precise measurement of $R_{K^{*0}}$ can provide a deeper understanding of the nature of the present discrepancies[15]. Some of the leading-order Feynman diagrams for the

$B^0 \to K^{*0} l^+ l^-$ decays, where l represents either a muon or an electron, are shown in Figure.1.3 for both Standard Model and possible New Physics scenarios. If the new physics particles couple differently to electrons and muons, lepton universality could be violated.

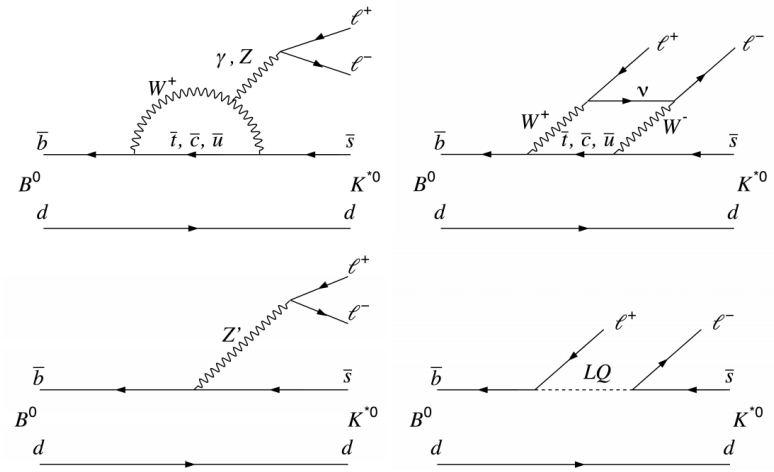

Figure 1.3: Feynman diagrams in the Standard Model of the $B^0 \to K^{*0} l^+ l^-$ decay for the (top left) electroweak penguin and (top right) box diagram. Possible new physics contribution violating Lepton Universality: (bottom left) a tree-level diagram mediated by a new gauge boson Z' and (bottom right) a tree-level diagram involving a leptoquark.[16]

Chapter 2

Method

2.1 Estimation of event number

The number of signal and background events in real data need to be estimated, so that the goodness of a given cut could be evaluated and finally the cuts could be optimized.

The formula to estimate the event number in real data is as follows:

$$N_{estimated} = \int Ldt * \sigma_{b\bar{b}} * \epsilon_{geo} * f_{hadron} * 2 * B_{decay}.$$

$\int Ldt$ is the integrated luminosity; $\sigma_{b\bar{b}}$ is the total production cross section of $b - \bar{b}$ quarks; ϵ_{geo} is the geometrical efficiency of a decay channel and this term is due to the geometrical structure of the detector; f_{hadron} is the hadronization factor of a certain hadron; the factor 2 is due to the charge conjugate process; B_{decay} refers to the branching fraction of a decay channel or decay chain which is composed of several decay processes.

2.2 Boosted Decision Tree

2.2.1 Decision tree

A decision tree is an extended cut-based selection. Many events do not have all characteristics of signal or background, and one should try not to rule out events failing a particular criterion. One should keep events rejected by one criterion and see whether other criteria could help classify them properly. Binary trees can be built with branches splitting into many sub-branches. It's visualization is like a flowchart

diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

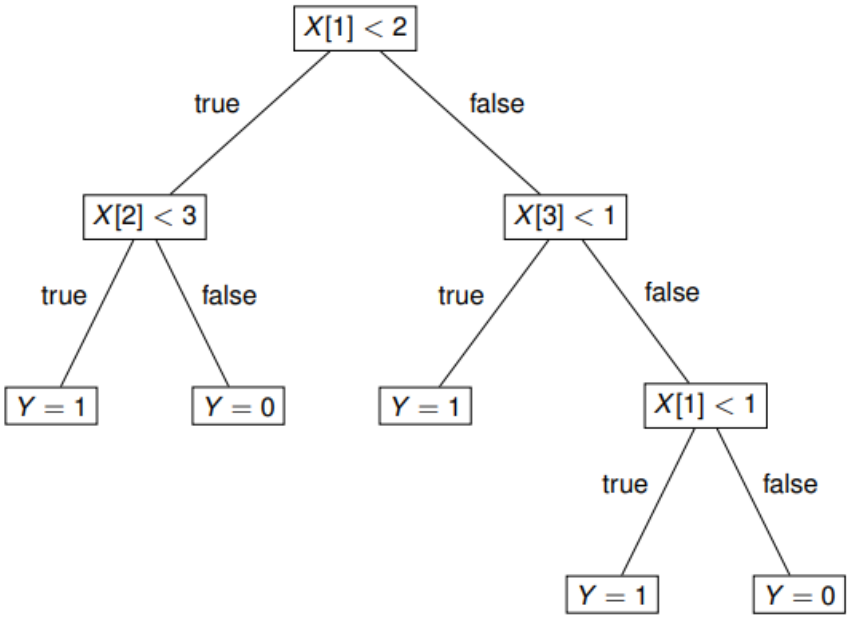

Figure 2.1: Decision tree[17]

The following is the procedure to grow a tree[18]:

- Start with setting all events(signal and background) in the first(root) node.
- Sort all events by each variable
- For each variable, find the splitting value with best separation between two children. Mostly signal in one child and background in the other.
- Select the variable and splitting value with best separation and produce two branches(nodes). The events failing criterion on one side and events passing it on the other.
- Keep splitting and now have two new nodes. Repeat algorithm recursively on each node. The same variable can be reused.
- Iterate until stopping criterion is reached.

- Splitting stops when the terminal node arrives at the final leaf.

The impurity measure should be made when splitting a node. The impurity measure $i(t)$ is maximal for equal mix of signal and background and it's symmetric in the probability p_{signal} and $p_{background}$. It's minimal for a node with either signal only or background only. The impurity measure is strictly concave. It rewards purer nodes or favors end cuts with one smaller node and one larger node. The decrease of impurity for split s of node t into children t_P and t_F (goodness of split) is defined by

$$\Delta i(s,t) = i(t) - p_P * i(t_P) - p_F * i(t_F).$$

The aim is to find the split s^* such that

$$\Delta i(s^*,t) = max_{s \in splits} \Delta i(s,t).$$

Maximizing $\Delta i(s,t)$ is just minimizing the overall tree impurity.

There are some common impurity functions:

- misclassification error $= 1 - max(p, 1-p)$
- (cross) entropy $= -\Sigma_{i=s,b} p_i log(p_i)$
- Gini index
- cross section $= -\frac{s^2}{s+b}$
- excess significance $= -\frac{s^2}{b}$

The Gini index of diversity is defined for many classes: $Gini = \Sigma^{i \neq j}_{i,j \in classes} p_i p_j$. Under statistical interpretation, if we assign a random object to class i with probability p_i and probability of class j is p_j, then $Gini = probability\ of\ misclassification$. For two classes like signal and background, $i = s, b$ and $p_s = p = 1 - p_b$, then $Gini = 1 - \Sigma_{i=s,b} p_i^2 = 2p(1-p) = \frac{2sb}{(s+b)^2}$.

The decision to stop splitting is made under the following conditions:

- Minimum leaf size is reached.
- Insufficient improvement from further splitting.
- Perfect classification(all events in leaf belong to same class)
- Maximal tree depth(like-size trees choice or computing concerns)

The number of variables is not affected too much by "curse of dimensionality" and the CPU consumption scales as $nNlogN$ with n variables and N training events. The result is insensitive to duplicate variables and the order of the variables doesn't matter. The order of training events is irrelevant. The irrelevant variables have no discriminative power and won't be used, and it only costs a little CPU time and there is no added noise. The continuous and discrete variables can be used simultaneously. The result is completely insensitive to the replacement of any subset of input variables by (possibly different) arbitrary strictly monotone functions of them. The ranking of one variable can be determined by adding up decrease of impurity each time the variable is used, and the largest decrease of impurity indicates the best variable.

There are some common tree parameters[19]:

- Max.depth: how tall a tree can grow.
- Max.features: how many features can be used to build a given tree.
- Min.samples per leaf: how many samples are required to make a new leaf.

These paremeters define the end condition for building a new tree. They are usually tuned to increase accuracy and prevent overfitting. The Max.depth is usually needed to be smaller than 10, and sometimes it's defined by number of leaves. The features are randomly selected from total set and the tree doesn't have to use all of the available features. The Min.samples per leaf is usually needed to be smaller than 1% of data. Sometimes it's defined by samples per split.

Ensemble learning, in general, is a model that makes predictions based on a number of different models[20][21]. By combining a number of different models, an ensemble learning tends to be more flexible(less bias) and less data sensitive(less variance).

The two most popular ensemble learning methods are bagging and boosting. Bagging is to train a bunch of models in parallel way, and each model learns from a random subset of the data. Boosting is to train a bunch of models sequentially, and each model learns from the mistakes of the previous model. The application of bagging is found in Random Forests. Random forests are a parallel combination of decision trees. Each tree is trained on random subset of the same data and the results from all trees are averaged to find the classification. The application of boosting is found in Gradient Boosting Decision Trees.

2.2.2 Boosting

Boosting is a general method and not limited to decision trees. It's hard to make a very good learner, but it's easy to make simple, error-prone ones(but it's still better

than random guessing). The goal of boosting is to combine such weak classifiers into a new more stable one with smaller error.

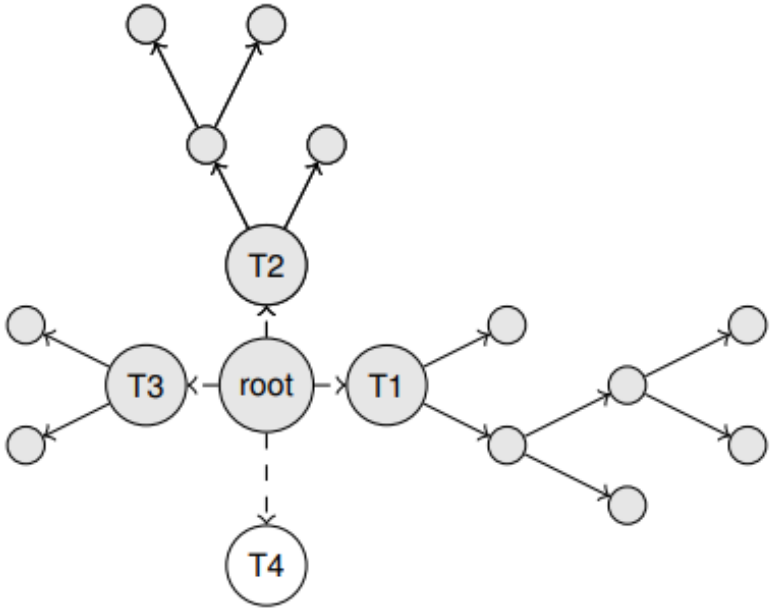

Figure 2.2: Ensemble of trees[17]

Boosting is a method of combining many weak learners(trees) into a strong classifier[19]. Usually each tree is created iteratively. The tree's output $h(x)$ is given a weight w relative to its accuracy, then the ensemble output is the weighted sum:

$$\hat{y}(x) = \Sigma_t w_t h_t(x).$$

After each iteration each data sample is given a weight based on its misclassification, and the more often a data sample is misclassified, the more important it becomes. The goal is to minimize an objective function:

$$O(x) = \Sigma_i l(\hat{y}_i, y_i) + \Sigma_t \Omega(f_t),$$

where $l(\hat{y}_i, y_i)$ is the loss function which stands for the distance between the truth and the prediction of the ith sample, $\Omega(f_t)$ is the regularization function which penalizes the complexity of the ith tree[22].

One of the very first boosting algorithms developed was Adaboost. The algorithm can be summarized as[18]:

- Initialize the first training sample
- Train the first classifier on the first training sample
- Train the Nth classifier on the Nth training sample
- assign weight to each Nth classifier
- Modify Nth classifier into next (N+1)th classifier
- Get boosted output from all the N classifiers and their weights

Gradient boosting algorithm is slightly different from Adaboost[23][24][25]. Instead of using the weighted average of individual outputs as the final outputs, it uses a loss function to minimize loss and converge upon a final output value. The loss function optimization is done using gradient descent, and hence the name gradient boosting. Further, gradient boosting uses short, less-complex decision trees instead of decision stumps.

All the trees are connected in series and each tree tries to minimise the error of the previous tree. Due to this sequential connection, boosting algorithms are usually slow to learn, but also highly accurate. In statistical learning, models that learn slowly perform better. The weak learners are fit in such a way that each new learner fits into the residuals of the previous step so as the model improves. The final model aggregates the result of each step and thus a strong learner is achieved.

Hyperparameters are key parts of learning algorithms which effect the performance and accuracy of a model. Learning rate and n-estimators are two critical hyperparameters for gradient boosting decision trees. Learning rate, denoted as α, simply means how fast the model learns. Each tree added modifies the overall model. The magnitude of the modification is controlled by learning rate.

There are some common boosting parameters[19]:

- Loss function: how to define the distance between the truth and the prediction.
- Learning rate: how much to adjust data weights after each iteration.
- Subsample size: How many samples to train each new tree.
- Number of trees: How many total trees to create.

The loss function can use binary logistic when there are two classes. Small learning rate is better but the training is slower. Usually the learning rate is somewhere around 0.1. Date samples are randomly selected each iteration. The number of trees is the same as the number of iteration, and usually more is better but may lead to overfitting.

The boosted decision tree has many benefits. Both training and prediction is fast, and the parameter is easy to tune. It's not sensitive to scale and the features can be a mix of categorical and continuous data. Training on the residuals gives very good accuracy so it has a good performance. Finally the boosted tree algorithms are very commonly used and there is a lot of well supported and well tested software available. There are also some problems about boosted decision tree. It's sensitive to overfitting and noise, therefore one should always crossvalidate. Modern software libraries have tools to avoid overfitting.

2.2.3 TMVA

The Toolkit for Multivariate Analysis(TMVA) is a powerful method which runs in a ROOT-integrated environment and can be used for the processing, parallel evaluation and application of multivariate classification[26]. The multivariate techniques of TMVA use supervised learning algorithms, and they make use of training events to determine the mapping function which could describes a decision boundary(classification) or an approximation of the underlying functional behavior defining the target value(regression). The function can be a single global function or a set of local models and can contain various degrees of approximations. The TMVA software package is composed of abstract, objected-oriented implementations in ROOT for each of these multivariate analysis techniques and other auxiliary tools, and it provides training, testing and evaluation algorithms and visualization scripts.

Chapter 3

Selection based on cuts

3.1 Preselection

3.1.1 Dataset

The samples used in this analysis are generated by simulation which correspond to the data in the year 2016. Three decay channels are studied:

- $B^0 \to K^+\pi^-e^+e^-$: the signal mode of this analysis.
- $B^0 \to K^*(\to K^+\pi^-)J/\psi(\to e^+e^-)$: the background channel to study e-h swap background.
- $\Lambda_b^0 \to pK^-J/\psi(\to e^+e^-)$: the background channel to study the background due to particle misidentification.

3.1.2 Preselection

The same preselection conditions need to be applied to all samples before further study.

The basic preselection is the requirement of the quality of tracks and particles. For tracks, the requirements are:

- $\chi^2/ndf < 3$
- $GhostProb < 0.4$

And the quality requirements of electrons are:

- the detection region in the Electromagnetic Calorimeter ≥ 0
- the projection to x-axis of the detection position in the Electromagnetic Calorimeter $> 363.6mm$ or the projection to y-axis $> 282.6mm$
- within the acceptance of the Electromagnetic Calorimeter

There are some preselections on particle identification:

- All particles: $hasRich$
- For electrons: $hasCalo$

The requirements on momentum are:

- For K, π: $p_T > 250MeV/c$
- For e: $p_T > 500MeV/c, p > 3000MeV/c$

And the requirements on the particles identification probabilities are:

- For K: $ProbNNk$[1]$*(1-ProbNNp$[2]$) > 0.05$, the difference of the log-likelihood between K and $\pi > 0$
- For π: $ProbNN\pi$[3]$ * (1 - ProbNNk) * (1 - ProbNNp) > 0.1$
- For e: $ProbNNe$[4]$ > 0.2$, the difference of the log-likelihood between e and $\pi > 2$

The angles between particles are restricted:

- $\theta(\pi, e) > 0.0005, \theta(K, e) > 0.0005, \theta(K, \pi) > 0.0005$

Since the region, where the invariant mass of K, π is smaller than $1GeV$, has been well studied, the requirement on the invariant mass of K, π is:

- $m(k\pi) > 1GeV$

Besides, there is requirement on q^2, the square of the invariant mass of the lepton pair. To avoid the influence from $\phi(1020) \to l^+l^-$ and from the J/ψ resonance, the requirement is:

- $1.1GeV < q^2 < 6.0GeV$

[1] the probability to be a Kaon calculated by neural network
[2] the probability to be a proton calculated by neural network
[3] the probability to be a pion calculated by neural network
[4] the probability to be an electron calculated by neural network

3.2 J/ψ e-h swap background

To optimize the cuts, the cuts efficiencies need to be determined, and then the signal event number s and background event number b of real data could be estimated. Then by comparing the significance values $\frac{s}{\sqrt{s+b}}$ of different cuts, the best cut can be determined which corresponds to the highest significance value.

The following are the constants that need to be used in the estimation:

- Luminosity of 2016: 1.6 fb^{-1}
- b quark total production cross section: 560 μb [27]
- B^0 hadronization fraction: 0.412 [28]
- $B^0 \to K^+\pi^- e^+ e^-$ branching ratio: 10.3 $\times 10^{-7}$ [29]
- detector geometric efficiency for $B^0 \to K^+\pi^- e^+ e^-$: 16.3223%
- $B^0 \to K^{*0} J/\psi$ branching ratio: 1.27 $\times 10^{-3}$ [29]
- $J/\psi \to e^+ e^-$ branching ratio: 5.971 $\times 10^{-2}$ [29]
- detector geometric efficiency for $B^0 \to K^{*0} J/\psi(\to e^+ e^-)$: 16.678%

As an estimation, the signal event number before multiplying by the total cuts efficiency is ≈ 124000 .

As an estimation, the background event number before multiplying by the total cuts efficiency is ≈ 9337500 .

3.2.1 K-e misidentification background

Two variables are used to veto this background:

- Jpsi_cons_mix_cov_double_kemisid_B_M: the invariant mass of $(K\pi ee)$ after swapping the K and the electron of the same charge.
- L1_MC15TuneV1_ProbNNe: the probability of the electron to be an electron.

The procedure to set cuts is: firstly, scan the lower boundary of Jpsi_cons_mix_cov_double_kemisid_B_M from 4500 to 4590 for 10 cut values: 4500, 4510, 4520, 4530, 4540, 4550, 4560, 4570, 4580, 4590; secondly, scan the upper boundary of Jpsi_cons_mix_cov_double_kemisid_B_M from 4601 to 5230.1 for 10 cut values: 4601, 4670.9, 4740.8,

4810.7, 4880.6, 4950.5, 5020.4, 5090.3, 5160.2, 5230.1; thirdly, scan the upper boundary of L1_MC15TuneV1_ProbNNe from 0.21 to 0.561 for 10 cut values: 0.21, 0.249, 0.288, 0.327, 0.366, 0.405, 0.444, 0.483, 0.522, 0.561. Under combination there are 1000 veto cuts.

For each cut, the total cut efficiencies of signal sample and background are calculated, and then the numbers of signal events s and background events b in real data are estimated. The significance values of all cuts are calculated and under comparison the best cut is determined. Figure 3.1 shows the comparison of these 1000 significance values, and Figure 3.2 shows the best cut in two-dimensional distributions in background and signal sample.

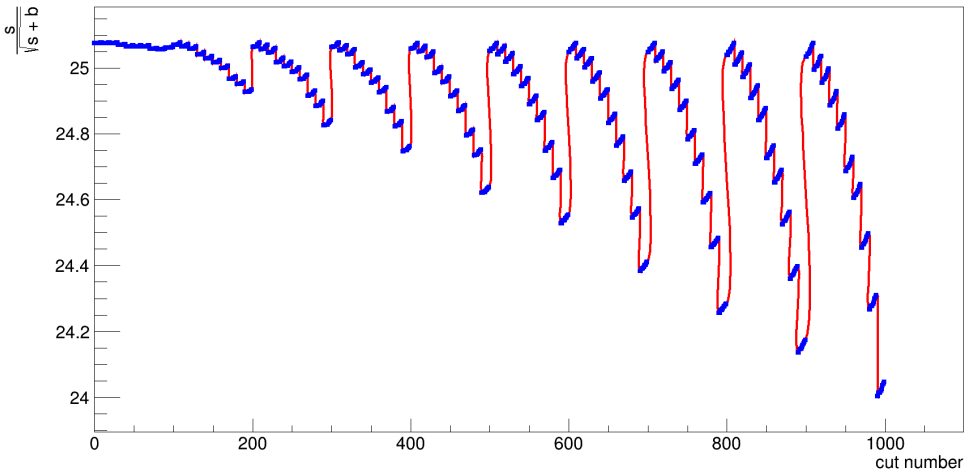

Figure 3.1: Plot $\frac{s}{\sqrt{s+b}}$ according to the order of cuts.

The best cut is: (4580 < Jpsi_cons_mix_cov_double_kemisid_B_M < 4601) & (L1_MC15TuneV1_ProbNNe < 0.21), and then s = 632.06, b = 3.18179, $\frac{s}{\sqrt{s+b}}$ = 25.0778.

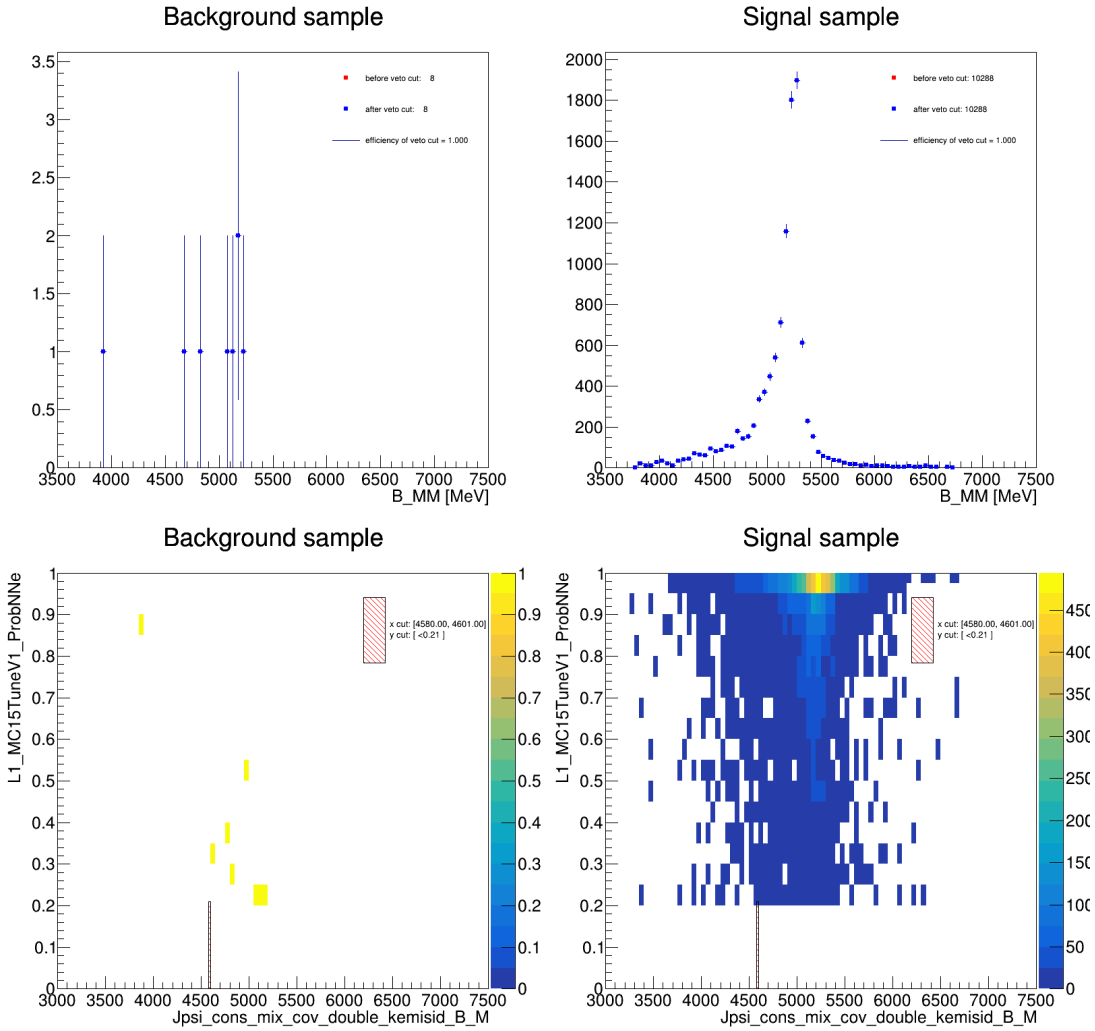

Figure 3.2: Top left: B mass distribution in background sample; top right: B mass distribution in signal sample; bottom left: veto cut on background sample; bottom right: veto cut on signal sample.

In Figure 3.2 the shaded rectangular region stands for the best cut. The region is rather small and it doesn't cut away any event. The main reason resulting in this situation is that there are very few background events.

3.2.2 π-e misidentification background

Two variables are used to veto this background:

- Jpsi_cons_mix_cov_double_piemisid_B_M: the invariant mass of $(K\pi ee)$ after swapping the π and the electron of the same charge.
- L2_MC15TuneV1_ProbNNe: the probability of the electron to be an electron.

The procedure to set cuts is: firstly, scan the lower boundary of Jpsi_cons_mix_cov_double_piemisid_B_M from 3500 to 3950 for 10 cut values: 3500, 3550, 3600, 3650, 3700, 3750, 3800, 3850, 3900, 3950; secondly, scan the upper boundary of Jpsi_cons_mix_cov_double_piemisid_B_M from 4001 to 4900.1 for 10 cut values: 4001, 4100.9, 4200.8, 4300.7, 4400.6, 4500.5, 4600.4, 4700.3, 4800.2, 4900.1; thirdly, scan the upper boundary of L2_MC15TuneV1_ProbNNe from 0.21 to 0.561 for 10 cut values: 0.21, 0.249, 0.288, 0.327, 0.366, 0.405, 0.444, 0.483, 0.522, 0.561. Under combination there are 1000 veto cuts.

For each cut, the total cut efficiencies of signal sample and background are calculated, and then the numbers of signal events s and background events b in real data are estimated. The significance values of all cuts are calculated and under comparison the best cut is determined. Figure 3.3 shows the comparison of these 1000 significance values, and Figure 3.4 shows the best cut in two-dimensional distributions in background and signal sample.

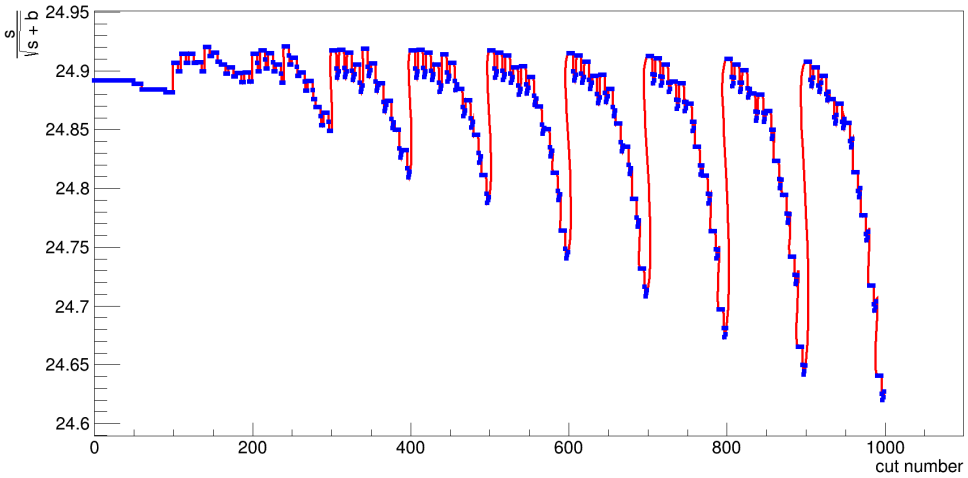

Figure 3.3: Plot $\frac{s}{\sqrt{s+b}}$ according to the order of cuts.

The best cut is: (3500 < Jpsi_cons_mix_cov_double_piemisid_B_M < 4400.6) & (L2_MC15TuneV1_ProbNNe < 0.288), and then s = 630.831, b = 9.9431, $\frac{s}{\sqrt{s+b}}$ = 24.9207.

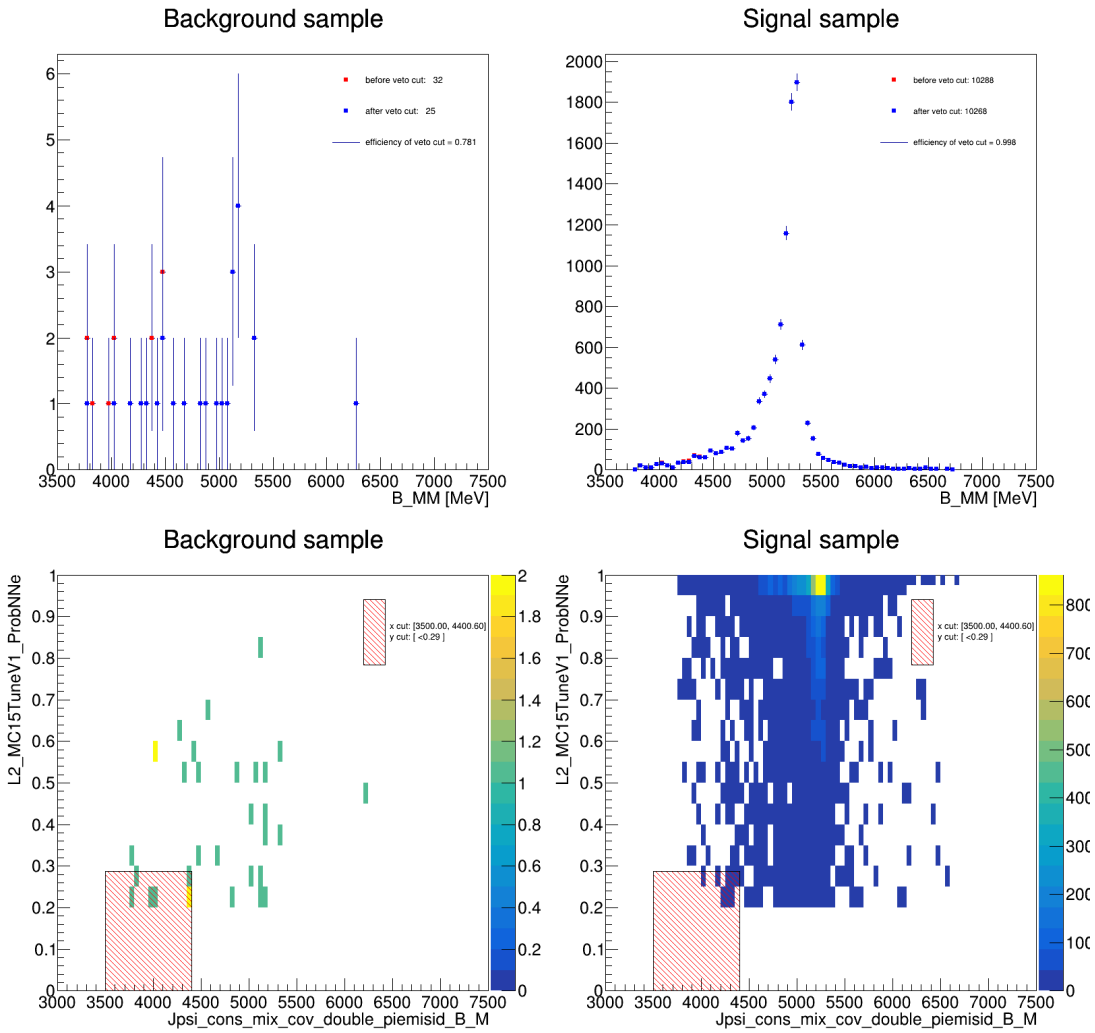

Figure 3.4: Top left: B mass distribution in background sample; top right: B mass distribution in signal sample; bottom left: veto cut on background sample; bottom right: veto cut on signal sample.

From Figure 3.4 it can be seen that the shaded rectangular region cuts away a small part of background events. These events have relatively small Jpsi_cons_mix_cov_double_piemisid_B_M values and small L2_MC15TuneV1_ProbNNe values.

3.3 $\Lambda_b^0 \to pK^- J/\psi(\to e^+e^-)$ background

The same procedures as above are performed on this background sample.

The following are the constants that need to be used in the estimation:

- Luminosity of 2016: 1.6 fb^{-1}
- b quark total production cross section: 560 μb [27]
- B^0 hadronization fraction: 0.412 [28]
- $B^0 \to K^+\pi^- e^+e^-$ branching ratio: 10.3 $\times 10^{-7}$ [29]
- detector geometric efficiency for $B^0 \to K^+\pi^- e^+e^-$: 16.3223%
- $\frac{\Lambda_b^0 hadronization fraction}{B^0 hadronization fraction}$: 0.387 [30]
- $\Lambda_b^0 \to pKJ/\psi$ branching ratio: 3.17 $\times 10^{-4}$ [29]
- $J/\psi \to e^+e^-$ branching ratio: 5.971 $\times 10^{-2}$ [29]
- detector geometric efficiency for $\Lambda_b^0 \to pKJ/\psi(\to e^+e^-)$: 17.3665%

As an estimation, the signal event number before multiplying by the total cuts efficiency is $\approx$ 124000 .

As an estimation, the background event number before multiplying by the total cuts efficiency is $\approx$ 939000 .

3.3.1 $\pi \to p$ misidentification background

Two variables are used to veto this background:

- B_M0123_Subst1_pi2p: the invariant mass of ($K\pi ee$) after substituting the π to a proton.
- Pi_PIDp: the difference of log-likelihood between π and proton.

The procedure to set cuts is: firstly, scan the lower boundary of B_M0123_Subst1_pi2p from 4000 to 4450 for 10 cut values: 4000, 4050, 4100, 4150, 4200, 4250, 4300, 4350, 4400, 4450; secondly, scan the upper boundary of B_M0123_Subst1_pi2p from 4501 to 5850.1 for 10 cut values: 4501, 4650.9, 4800.8, 4950.7, 5100.6, 5250.5, 5400.4, 5550.3, 5700.2, 5850.1; thirdly, scan the lower boundary of Pi_PIDp from -20 to 16

for 10 cut values: -20, -16, -12, -8, -4, 0, 4, 8, 12, 16. Under combination there are 1000 veto cuts.

For each cut, the total cut efficiencies of signal sample and background are calculated, and then the numbers of signal events s and background events b in real data are estimated. The significance values of all cuts are calculated and under comparison the best cut is determined. Figure 3.5 shows the comparison of these 1000 significance values, and Figure 3.6 shows the best cut in two-dimensional distributions in background and signal sample.

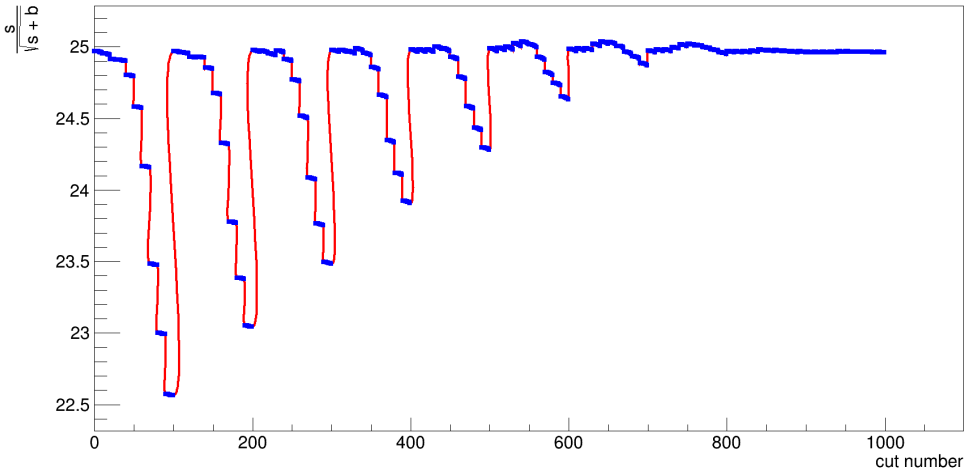

Figure 3.5: Plot $\frac{s}{\sqrt{s+b}}$ according to the order of cuts.

The best cut is: (4000<B_M0123_Subst1_pi2p<5100.6) & (Pi_PIDp>0), and then s = 629.971, b = 3.07363, $\frac{s}{\sqrt{s+b}}$ = 25.0382.

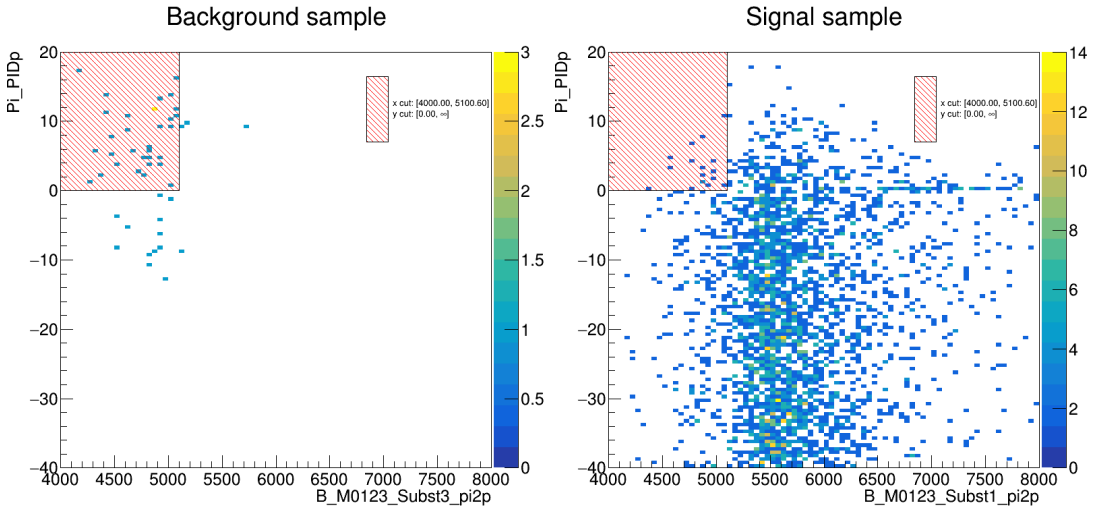

Figure 3.6: Left: veto cut on background sample; right: veto cut on signal sample.

From Figure 3.6 it can be seen that the background events and signal events don't overlap each other too much, thus the shaded rectangular region cuts away a large part of background events. These events have relatively large Pi_PIDp values.

3.3.2 $K\pi \to pK$ misidentification background

Two variables are used to veto this background:

- B_M0123_Subst01_Kpi2pK: the invariant mass of $(K\pi ee)$ after substituting the K to a proton and the π to a K.
- Pi_PIDK: the difference of log-likelihood between π and K.

The procedure to set cuts is: firstly, scan the lower boundary of B_M0123_Subst01-_Kpi2pK from 4000 to 4495 for 10 cut values: 4000, 4055, 4110, 4165, 4220, 4275, 4330, 4385, 4440, 4495; secondly, scan the upper boundary of B_M0123_Subst01_K-pi2pK from 4551 to 5405.1 for 10 cut values: 4551, 4645.9, 4740.8, 4835.7, 4930.6, 5025.5, 5120.4, 5215.3, 5310.2, 5405.1; thirdly, scan the lower boundary of Pi_PIDK from -14 to 11.2 for 10 cut values: -14, -11.2, -8.4, -5.6, -2.8, 0, 2.8, 5.6, 8.4, 11.2. Under combination there are 1000 veto cuts.

For each cut, the total cut efficiencies of signal sample and background are calculated, and then the numbers of signal events s and background events b in real data are estimated. The significance values of all cuts are calculated and under comparison

the best cut is determined. Figure 3.7 shows the comparison of these 1000 significance values, and Figure 3.8 shows the best cut in two-dimensional distributions in background and signal sample.

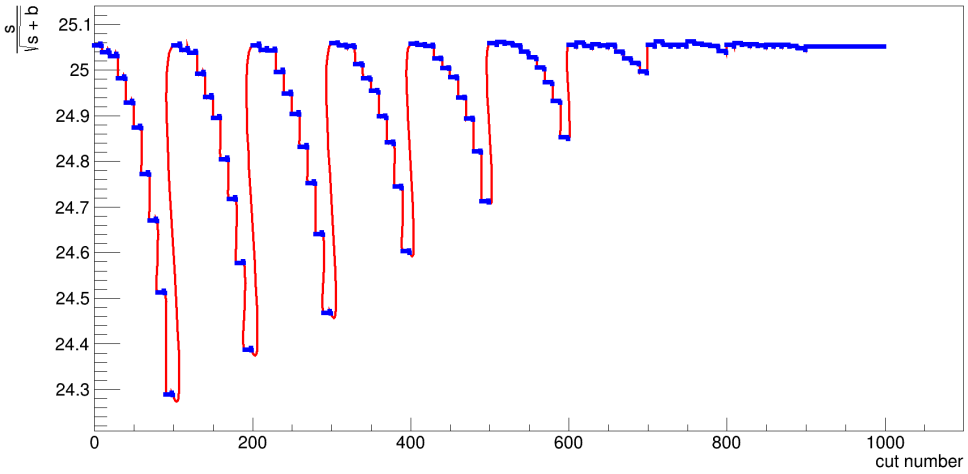

Figure 3.7: Plot $\frac{s}{\sqrt{s+b}}$ according to the order of cuts.

The best cut is: (4000<B_M0123_Subst01_Kpi2pK<5025.5) & (Pi_PIDK>5.6), and then s = 631.2, b = 3.07363, $\frac{s}{\sqrt{s+b}}$ = 25.0627.

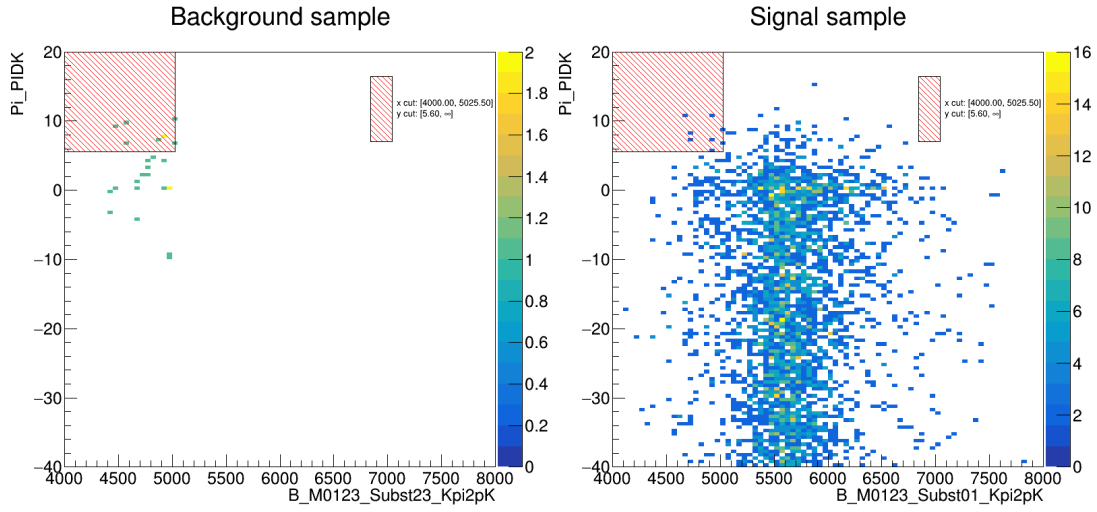

Figure 3.8: Left: veto cut on background sample; right: veto cut on signal sample.

From Figure 3.8 it can be seen that the background events are concentrated. The shaded rectangular region cuts away a small part of background events, which have relatively large Pi_PIDK values.

Chapter 4

Selection using Boosted Decision Tree

Now the Boosted Decision Tree method is used to optimize the selection. The same preselection cuts are applied before BDT selection but the particle identification cuts need to be removed from the preselection. After preselection there are 13964 signal events.

4.1 J/ψ e-h swap background

4.1.1 K-e misidentification background

For the K-e misidentification background, after preselection there are 372 events, and among them 240 events are used for training and 132 events for testing. Among 13964 signal events, 10000 events are used for training and 3964 events for testing.

Eight input variables are chosen for the training. They are: K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), L1_MC15TuneV1_ProbNNe, L2_MC15TuneV1_ProbNNe, Jpsi_cons_mix_cov_double_kemisid_B_M, L2_PIDe, K_PIDK, Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), L1_PIDe. Figure4.1 shows their distributions in signal sample and background sample. The overtraining check of the results is shown in Figure4.2 and the ROC curve is shown in Figure4.3.

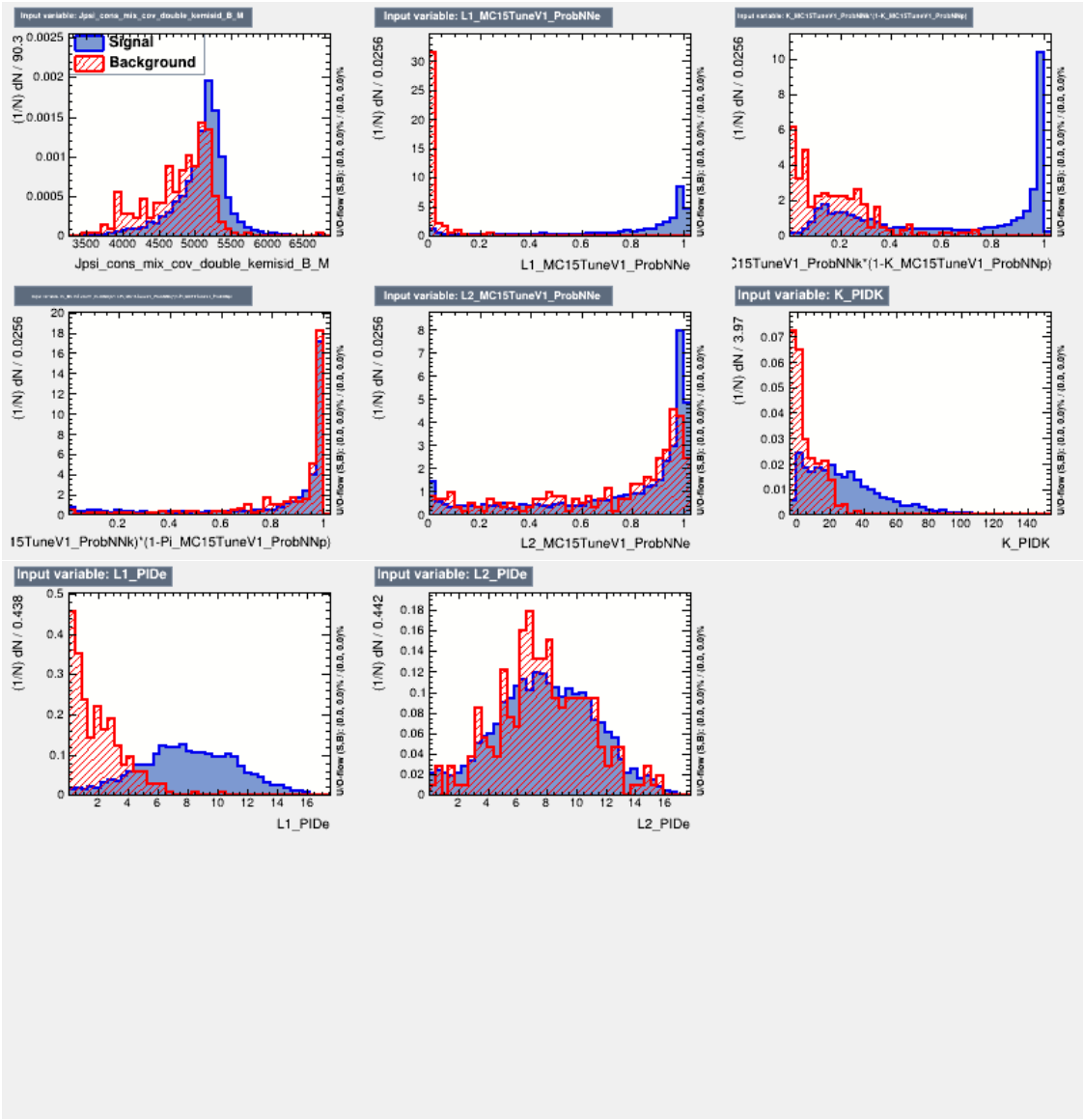

Figure 4.1: The distributions of input variables in signal sample and background sample.

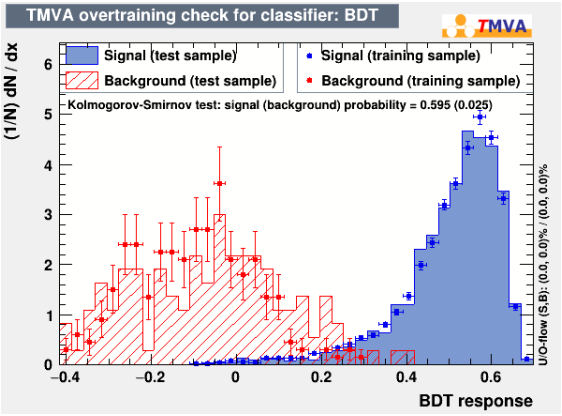

Figure 4.2: The overtraining check for classifier.

From the figure above it can be seen that there is no obvious overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.888 and 0.954; when background efficiency is 0.10, the signal efficiencies from test and training sample are 0.978 and 0.988; when background efficiency is 0.30, the signal efficiencies from test and training sample are 0.991 and 1.000.

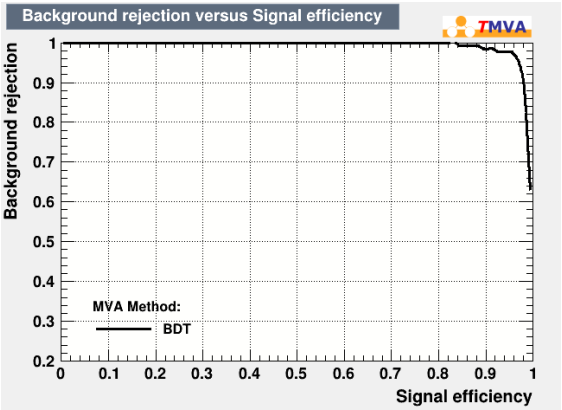

Figure 4.3: The ROC curve of BDT method.

The area integral under the ROC curve is 0.991.

Now reduce the number of input variables to five and see the change of the results.

The second set of input variables used for the training are: K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), L1_MC15TuneV1_ProbNNe, L2_MC15TuneV-

1_ProbNNe, Jpsi_cons_mix_cov_double_kemisid_B_M, L2_PIDe. The overtraining check of the results is shown in Figure4.4 and the ROC curve is shown in Figure4.5.

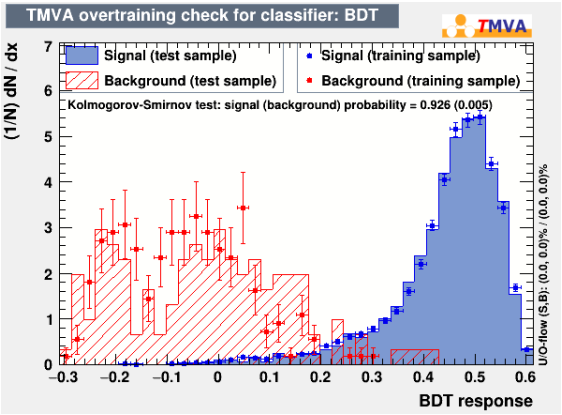

Figure 4.4: The overtraining check for classifier.

From the figure above it can be seen that there is a little overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.768 and 0.949; when background efficiency is 0.10, the signal efficiencies from test and training sample are 0.972 and 0.986; when background efficiency is 0.30, the signal efficiencies from test and training sample are 0.988 and 0.994.

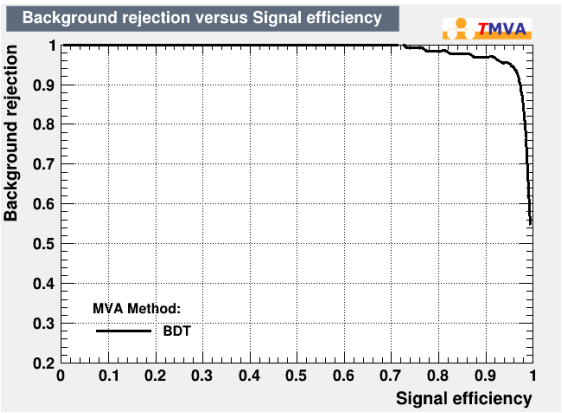

Figure 4.5: The ROC curve of BDT method.

The area integral under the ROC curve is 0.985.

Now choose another five input varialbes and see the change of the results.

The third set of input variables used for the training are: K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), Jpsi_cons_mix_cov_double_kemisid_B_M, L1_-MC15TuneV1_ProbNNe, K_PIDK, L1_PIDe. The overtraining check of the results is shown in Figure4.6 and the ROC curve is shown in Figure4.7.

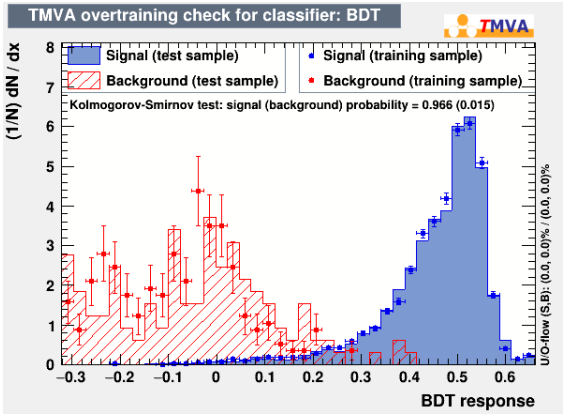

Figure 4.6: The overtraining check for classifier.

From the figure above it can be seen that there is no obvious overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.837 and 0.956; when background efficiency is 0.10, the signal efficiencies from test and training sample are 0.974 and 0.986; when background efficiency is 0.30, the signal efficiencies from test and training sample are 0.990 and 0.997.

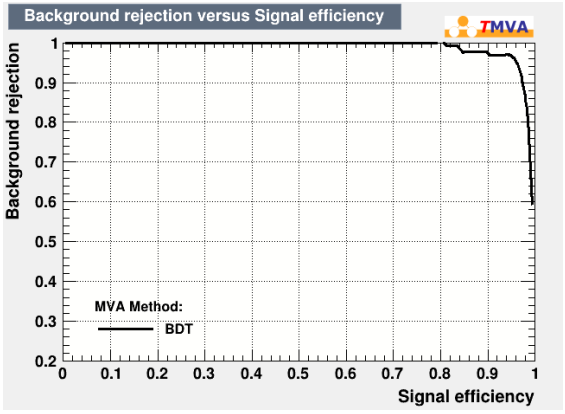

Figure 4.7: The ROC curve of BDT method.

The area integral under the ROC curve is 0.989.

Now reduce the number of input variables to three and see the change of the results.

The fourth set of input variables used for the training are: L2_PIDe, K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), L1_MC15TuneV1_ProbNNe. The overtraining check of the results is shown in Figure4.8 and the ROC curve is shown in Figure4.9.

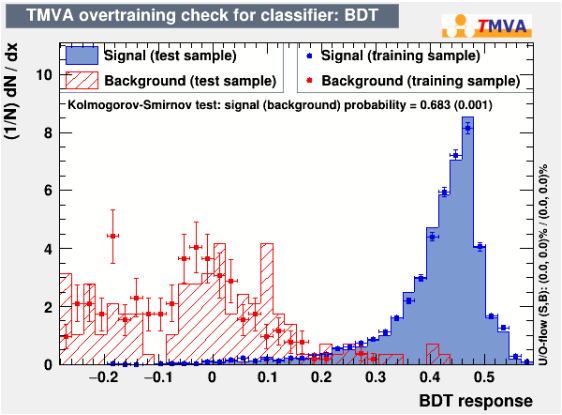

Figure 4.8: The overtraining check for classifier.

From the figure above it can be seen that the training is not good. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.618 and 0.929; when background efficiency is 0.10, the signal efficiencies from test and training sample are 0.969 and 0.985; when background efficiency is 0.30, the signal efficiencies from test and training sample are 0.987 and 0.994.

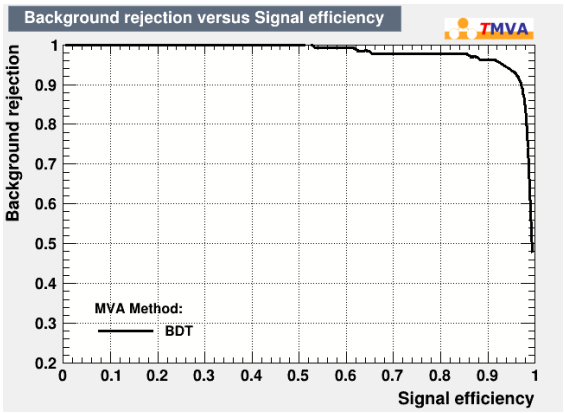

Figure 4.9: The ROC curve of BDT method.

The area integral under the ROC curve is 0.980.

4.1.2 π-e misidentification background

For the π-e misidentification background, after preselection there are 491 events, and among them 320 events are used for training and 171 events for testing. Among 13964 signal events, 9000 events are used for training and 4964 events for testing.

Eight input variables are chosen for the training. They are: Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), K_PIDK, L1_PIDe, L2_MC15TuneV1_ProbNNe, K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), Jpsi_cons_mix_cov_double_piemisid_B_M, L1_MC15TuneV1_ProbNNe, L2_PIDe. Figure4.10 shows their distributions in signal sample and background sample. The overtraining check of the results is shown in Figure4.11 and the ROC curve is shown in Figure4.12.

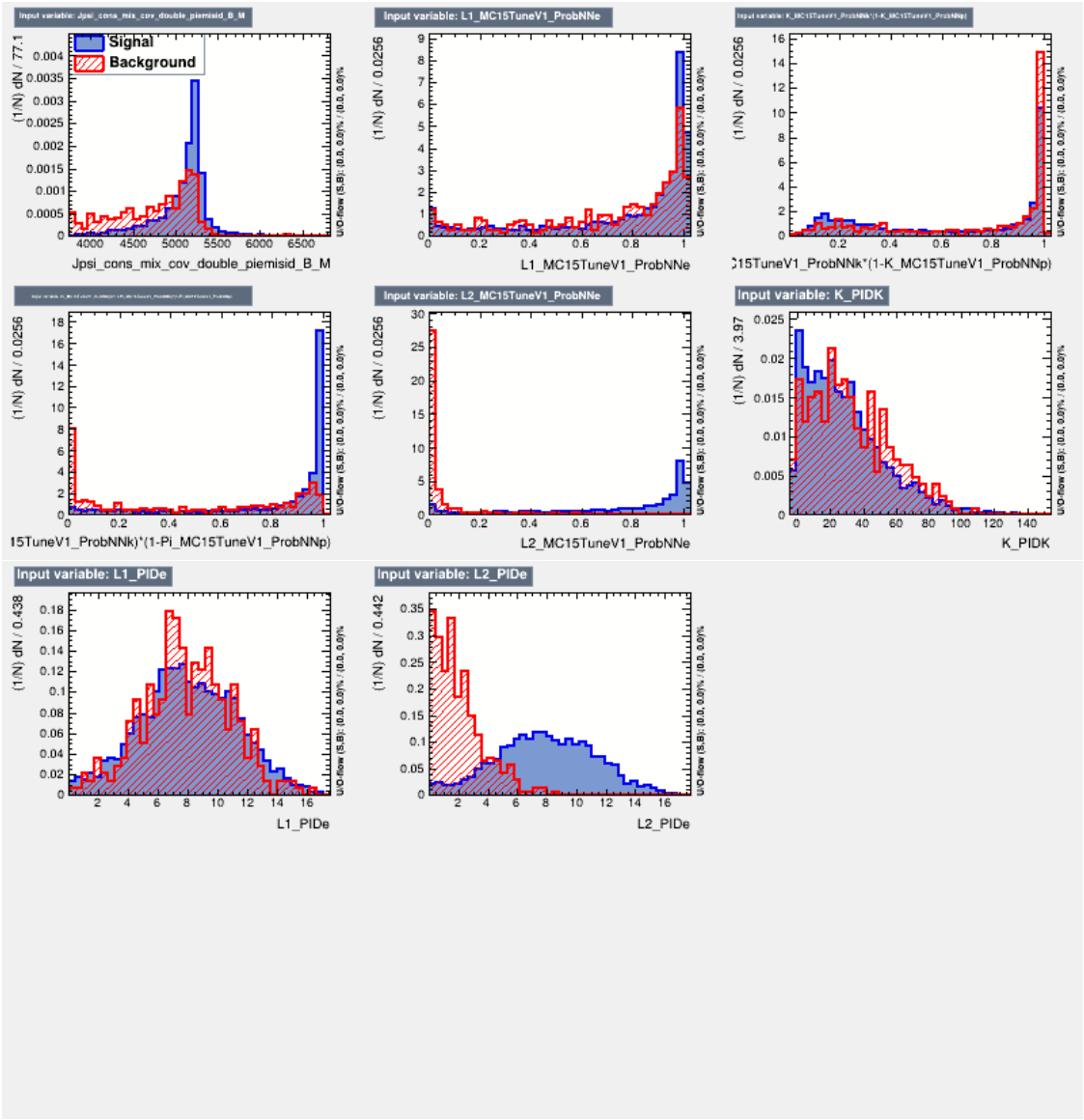

Figure 4.10: The distributions of input variables in signal sample and background sample.

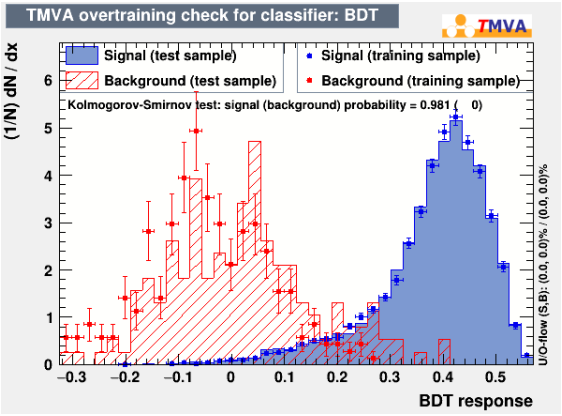

Figure 4.11: The overtraining check for classifier.

From the figure above it can be seen that there is a little overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.542 and 0.901; when background efficiency is 0.10, the signal efficiencies from test and training sample are 0.919 and 0.971; when background efficiency is 0.30, the signal efficiencies from test and training sample are 0.981 and 0.991.

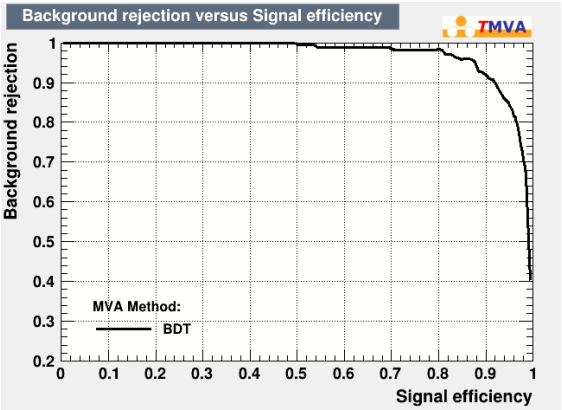

Figure 4.12: The ROC curve of BDT method.

The area integral under the ROC curve is 0.970.

Now reduce the number of input variables to five and see the change of the results.

The second set of input variables used for the training are: Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), K_PIDK,

L1_PIDe, L2_MC15TuneV1_ProbNNe, K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp). The overtraining check of the results is shown in Figure4.13 and the ROC curve is shown in Figure4.14.

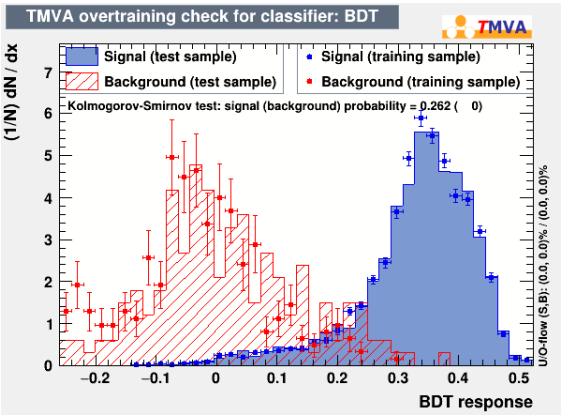

Figure 4.13: The overtraining check for classifier.

From the figure above it can be seen that there is no obvious overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.672 and 0.877; when background efficiency is 0.10, the signal efficiencies from test and training sample are 0.906 and 0.960; when background efficiency is 0.30, the signal efficiencies from test and training sample are 0.973 and 0.989.

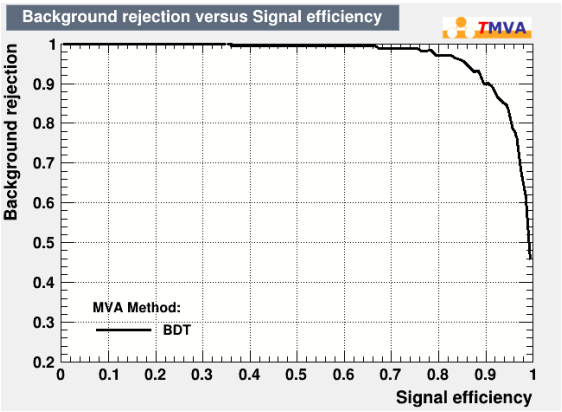

Figure 4.14: The ROC curve of BDT method.

The area integral under the ROC curve is 0.967.

Now choose another five input varialbes and see the change of the results.

The third set of input variables used for the training are: Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), L1_MC15TuneV1_ProbNNe, Jpsi_cons_mix_cov_double_piemisid_B_M, L2_MC15TuneV1_ProbNNe, L2_PIDe. The overtraining check of the results is shown in Figure4.15 and the ROC curve is shown in Figure4.16.

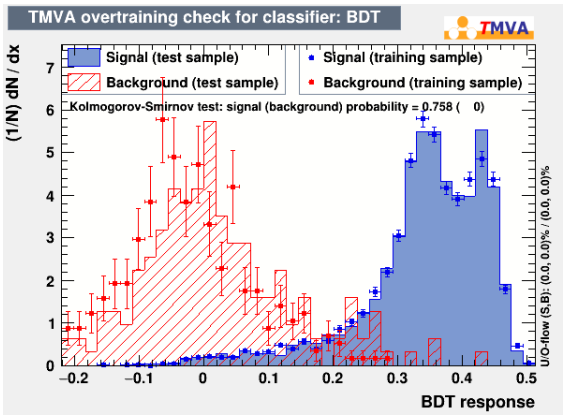

Figure 4.15: The overtraining check for classifier.

From the figure above it can be seen that there is a little overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.502 and 0.877; when background efficiency is 0.10, the signal efficiencies from test and training sample are 0.906 and 0.960; when background efficiency is 0.30, the signal efficiencies from test and training sample are 0.976 and 0.985.

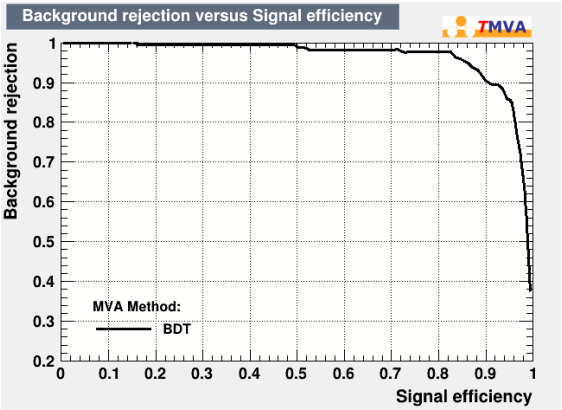

Figure 4.16: The ROC curve of BDT method.

The area integral under the ROC curve is 0.964.

Now reduce the number of input variables to three and see the change of the results.

The fourth set of input variables used for the training are: Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), K_PIDK, L1_PIDe. The overtraining check of the results is shown in Figure4.17 and the ROC curve is shown in Figure4.18.

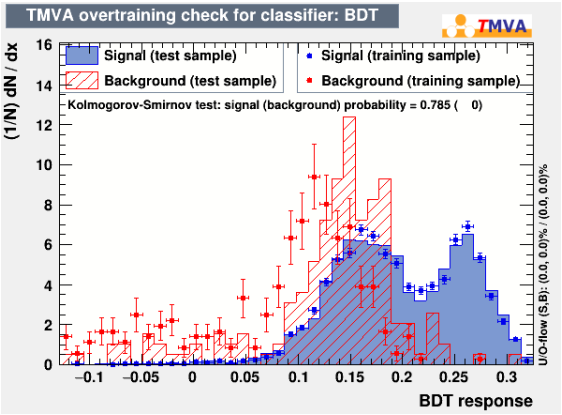

Figure 4.17: The overtraining check for classifier.

From the figure above it can be seen that the training is very bad. When background efficiency is 0.01, the signal efficiencies from test and training sample are

0.142 and 0.429; when background efficiency is 0.10, the signal efficiencies from test and training sample are 0.516 and 0.685; when background efficiency is 0.30, the signal efficiencies from test and training sample are 0.657 and 0.878.

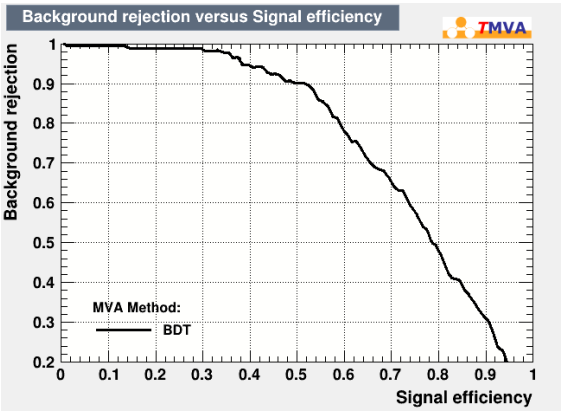

Figure 4.18: The ROC curve of BDT method.

The area integral under the ROC curve is 0.760.

4.2 $\Lambda_b^0 \to pK^- J/\psi(\to e^+e^-)$ background

4.2.1 $\pi \to p$ misidentification background

For the $\pi \to p$ misidentification background, after preselection there are 881 events, and among them 600 events are used for training and 281 events for testing. Among 13964 signal events, 9000 events are used for training and 4964 events for testing.

Nine input variables are chosen for the training. They are: Pi_PIDp, B_M0123_-Subst1_pi2p, Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_M-C15TuneV1_ProbNNp), K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), L2_MC15TuneV1_ProbNNe, L2_PIDe, L1_MC15TuneV1_ProbNNe, L1_PIDe, K_PIDK. Figure4.19 shows their distributions in signal sample and background sample. The overtraining check of the results is shown in Figure4.20 and the ROC curve is shown in Figure4.21.

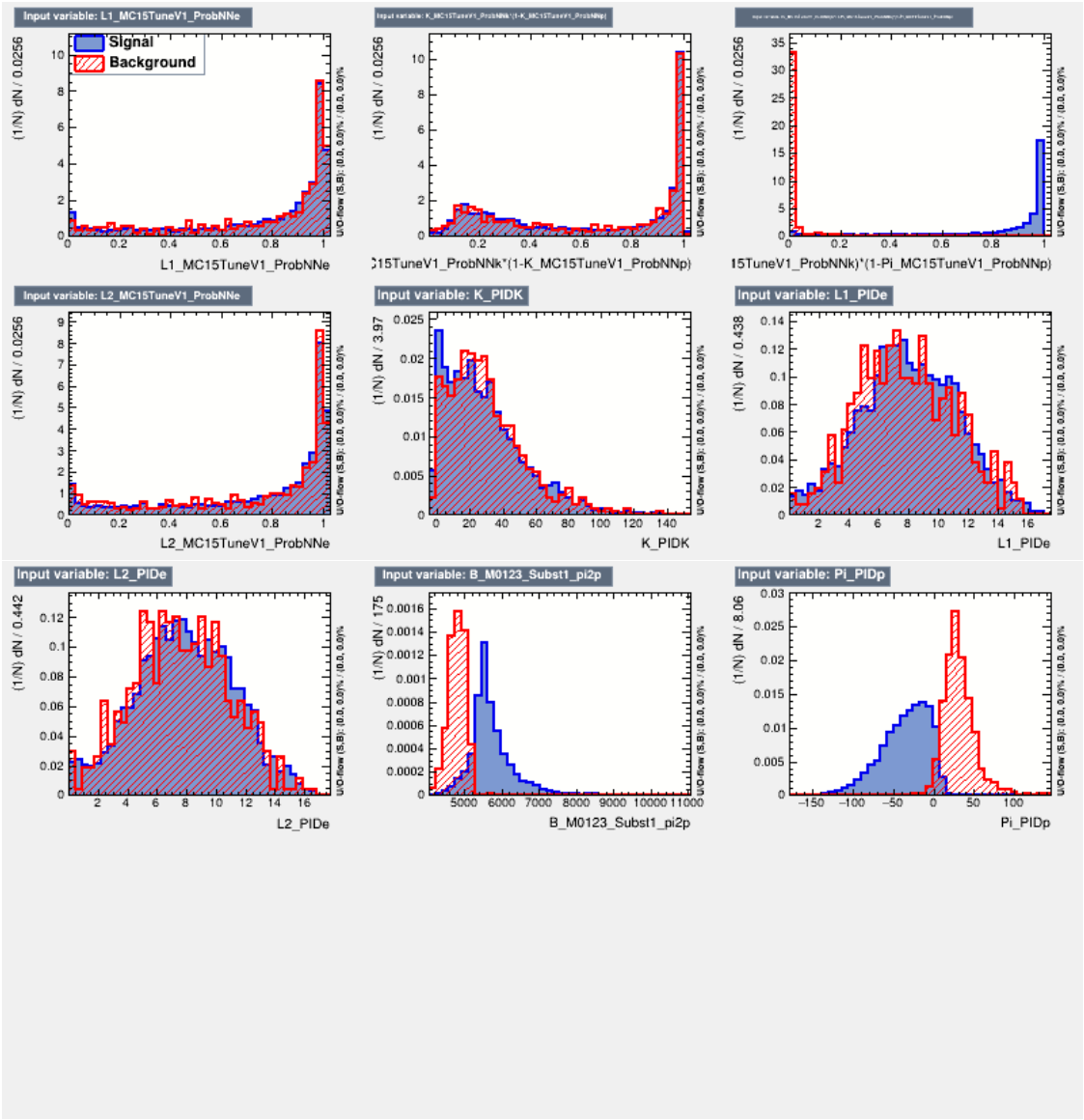

Figure 4.19: The distributions of input variables in signal sample and background sample.

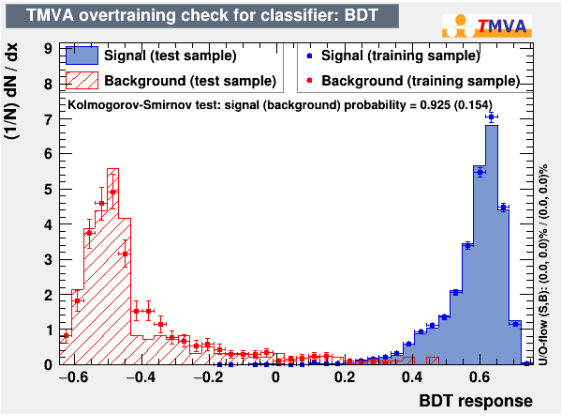

Figure 4.20: The overtraining check for classifier.

From the figure above it can be seen that there is no obvious overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.953 and 0.995; when background efficiency is 0.10, the signal efficiencies from test and training sample are 1.000 and 1.000; when background efficiency is 0.30, the signal efficiencies from test and training sample are 1.000 and 1.000.

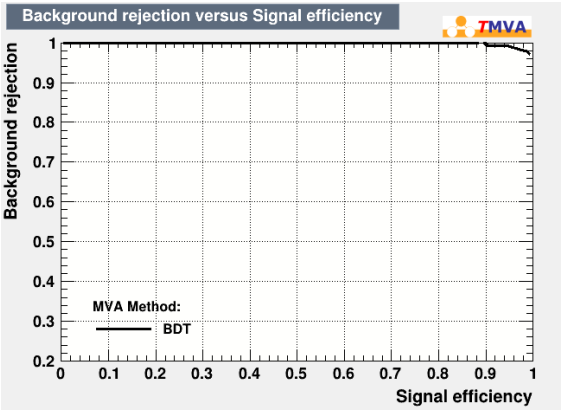

Figure 4.21: The ROC curve of BDT method.

The area integral under the ROC curve is 0.999.

Now reduce the number of input variables to five and see the change of the results.

The second set of input variables used for the training are: Pi_PIDp, B_M0123_-Subst1_pi2p, Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_M-

C15TuneV1_ProbNNp), K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), L2_MC15TuneV1_ProbNNe. The overtraining check of the results is shown in Figure4.22 and the ROC curve is shown in Figure4.23.

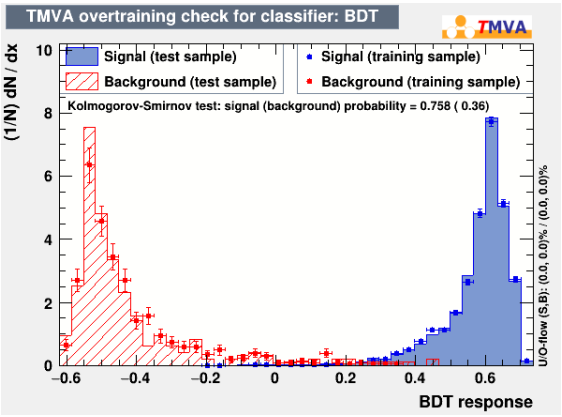

Figure 4.22: The overtraining check for classifier.

From the figure above it can be seen that there is no obvious overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.959 and 0.995; when background efficiency is 0.10, the signal efficiencies from test and training sample are 1.000 and 1.000; when background efficiency is 0.30, the signal efficiencies from test and training sample are 1.000 and 1.000.

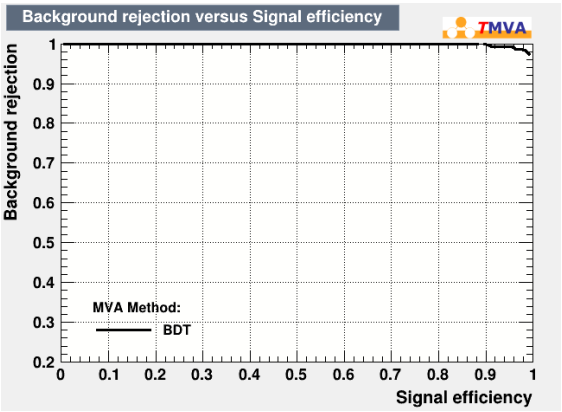

Figure 4.23: The ROC curve of BDT method.

The area integral under the ROC curve is 0.999.

Now choose another five input varialbes and see the change of the results.

The third set of input variables used for the training are: Pi_PIDp, Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), L2_PIDe, L1_PIDe, B_M0123_Subst1_pi2p. The overtraining check of the results is shown in Figure4.24 and the ROC curve is shown in Figure4.25.

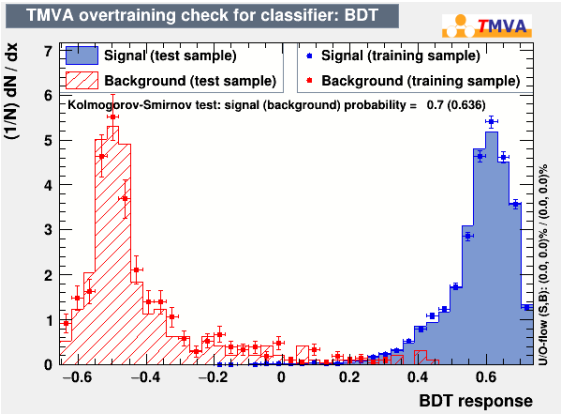

Figure 4.24: The overtraining check for classifier.

From the figure above it can be seen that there is no obvious overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.941 and 0.993; when background efficiency is 0.10, the signal efficiencies from test and training sample are 1.000 and 1.000; when background efficiency is 0.30, the signal efficiencies from test and training sample are 1.000 and 1.000.

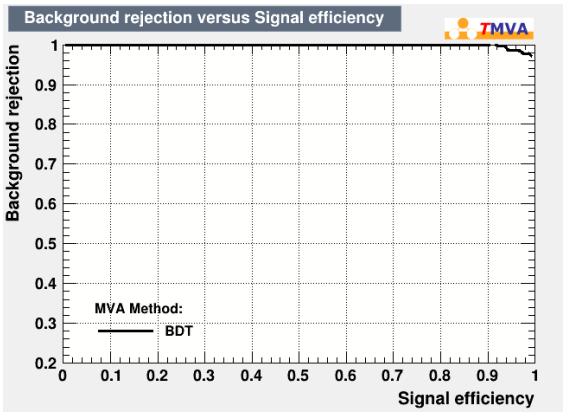

Figure 4.25: The ROC curve of BDT method.

The area integral under the ROC curve is 0.999.

Now reduce the number of input variables to three and see the change of the results.

The fourth set of input variables used for the training are: K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), Pi_PIDp. The overtraining check of the results is shown in Figure4.26 and the ROC curve is shown in Figure4.27.

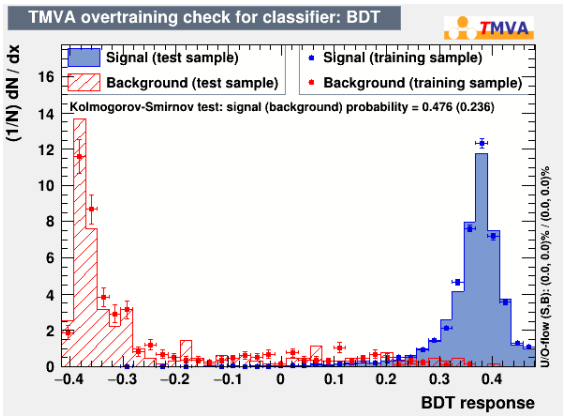

Figure 4.26: The overtraining check for classifier.

From the figure above it can be seen that the training is not so good. When background efficiency is 0.01, the signal efficiencies from test and training sample

are 0.763 and 0.945; when background efficiency is 0.10, the signal efficiencies from test and training sample are 0.992 and 0.995; when background efficiency is 0.30, the signal efficiencies from test and training sample are 1.000 and 1.000.

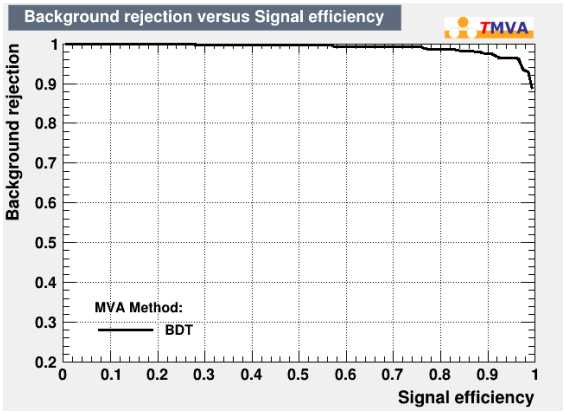

Figure 4.27: The ROC curve of BDT method.

The area integral under the ROC curve is 0.990.

4.2.2 $K\pi \to pK$ misidentification background

For the $K\pi \to pK$ misidentification background, after preselection there are 1018 events, and among them 600 events are used for training and 418 events for testing. Among 13964 signal events, 9000 events are used for training and 4964 events for testing.

Nine input variables are chosen for the training. They are: K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), Pi_PIDK, B_M0123_Subst01_Kpi2pK, Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), K_PIDK, L2_PIDe, L1_MC15TuneV1_ProbNNe, L1_PIDe, L2_MC15TuneV1_ProbNNe. Figure4.28 shows their distributions in signal sample and background sample. The overtraining check of the results is shown in Figure4.29 and the ROC curve is shown in Figure4.30.

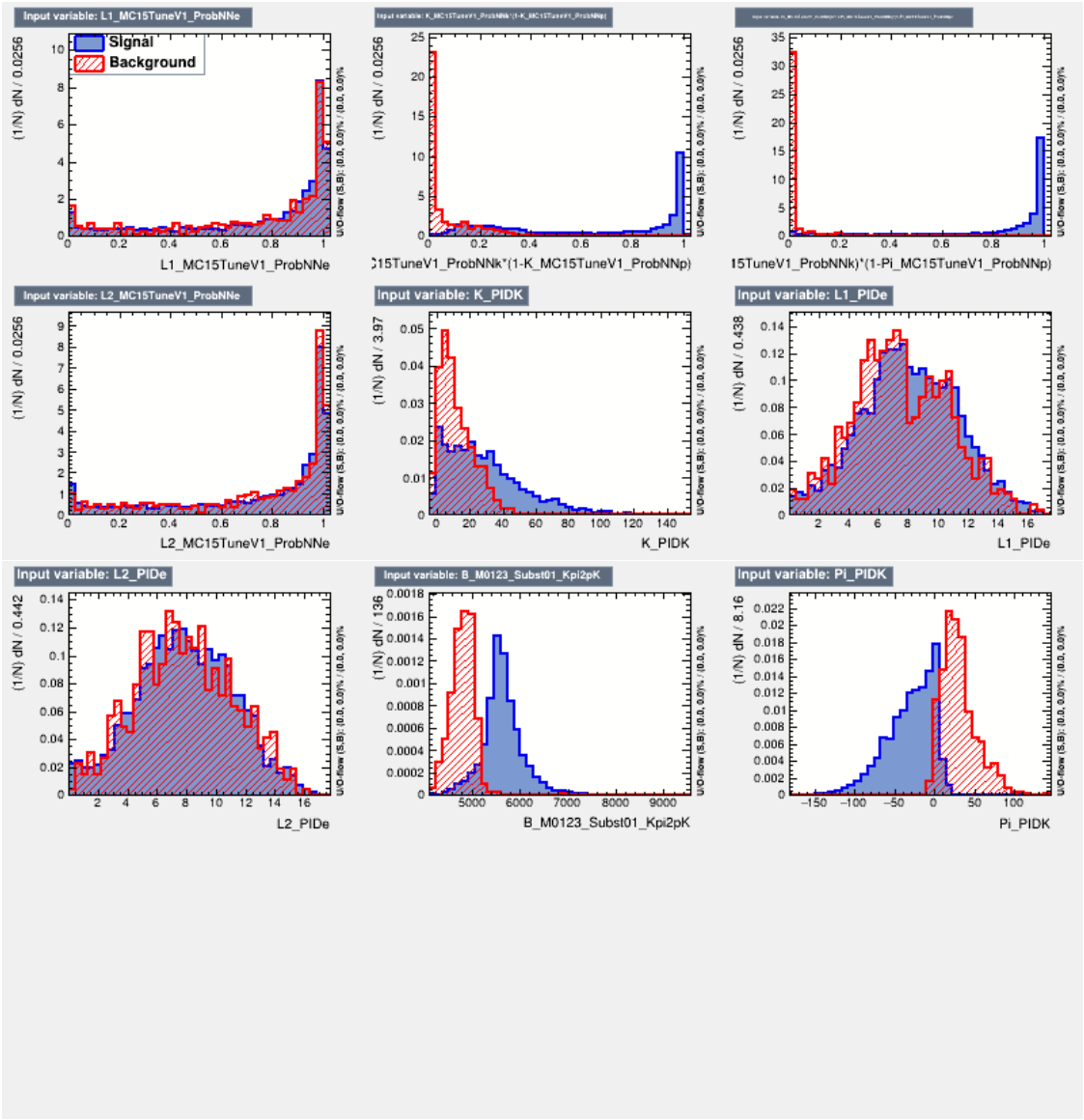

Figure 4.28: The distributions of input variables in signal sample and background sample.

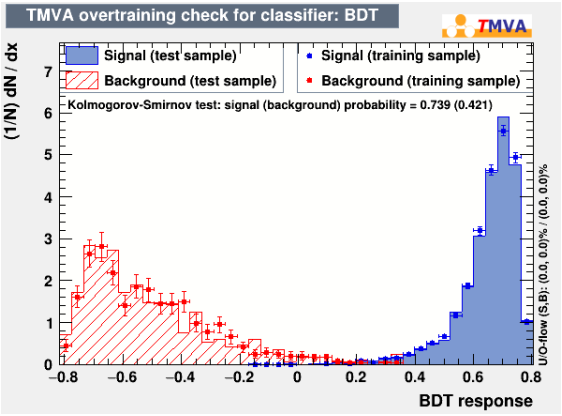

Figure 4.29: The overtraining check for classifier.

From the figure above it can be seen that there is no obvious overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.983 and 1.000; when background efficiency is 0.10, the signal efficiencies from test and training sample are 1.000 and 1.000; when background efficiency is 0.30, the signal efficiencies from test and training sample are 1.000 and 1.000.

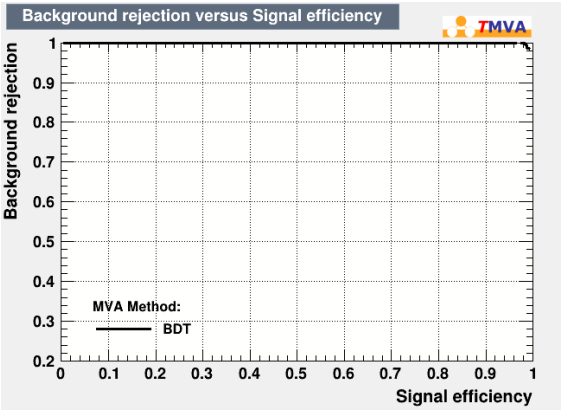

Figure 4.30: The ROC curve of BDT method.

The area integral under the ROC curve is 1.000.

Now reduce the number of input variables to five and see the change of the results.

The second set of input variables used for the training are: K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), Pi_PIDK, B_M0123_Subst01_Kpi2pK, Pi_MC-

15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), K_PIDK. The overtraining check of the results is shown in Figure4.31 and the ROC curve is shown in Figure4.32.

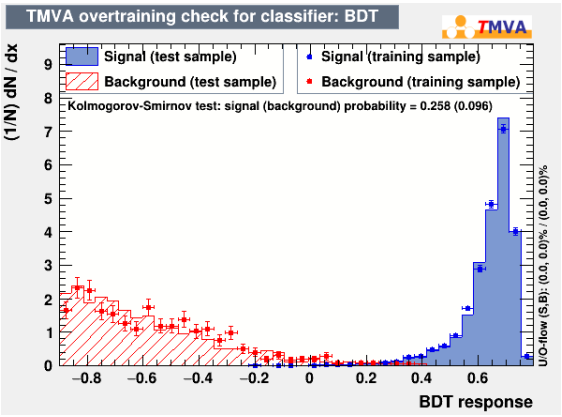

Figure 4.31: The overtraining check for classifier.

From the figure above it can be seen that there is no obvious overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.991 and 1.000; when background efficiency is 0.10, the signal efficiencies from test and training sample are 1.000 and 1.000; when background efficiency is 0.30, the signal efficiencies from test and training sample are 1.000 and 1.000.

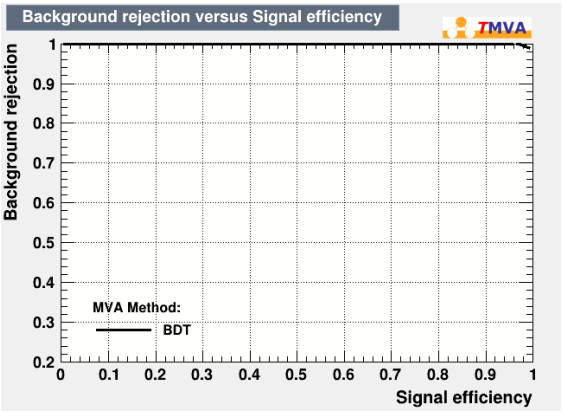

Figure 4.32: The ROC curve of BDT method.

The area integral under the ROC curve is 1.000.

Now choose another five input varialbes and see the change of the results.

The third set of input variables used for the training are: K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), B_M0123_Subst01_Kpi2pK, Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), Pi_PIDK, K_PIDK. The overtraining check of the results is shown in Figure4.33 and the ROC curve is shown in Figure4.34.

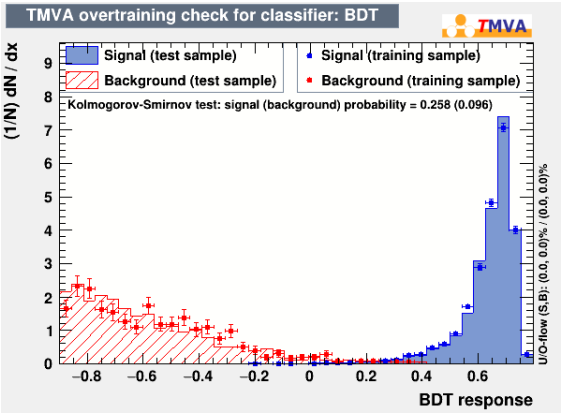

Figure 4.33: The overtraining check for classifier.

From the figure above it can be seen that there is no obvious overtraining. When background efficiency is 0.01, the signal efficiencies from test and training sample are 0.991 and 1.000; when background efficiency is 0.10, the signal efficiencies from test and training sample are 1.000 and 1.000; when background efficiency is 0.30, the signal efficiencies from test and training sample are 1.000 and 1.000.

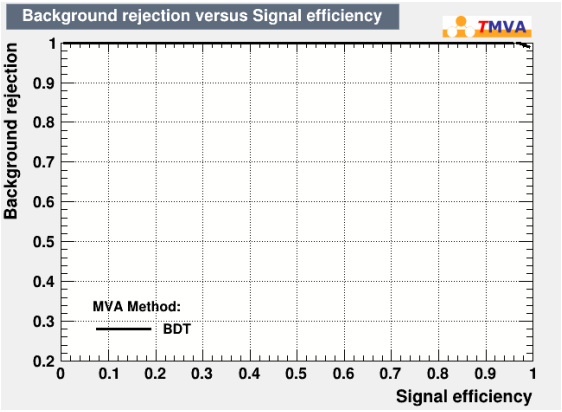

Figure 4.34: The ROC curve of BDT method.

The area integral under the ROC curve is 1.000.

Now reduce the number of input variables to three and see the change of the results.

The fourth set of input variables used for the training are: Pi_MC15TuneV1_ProbNNpi*(1-Pi_MC15TuneV1_ProbNNk)*(1-Pi_MC15TuneV1_ProbNNp), K_MC15TuneV1_ProbNNk*(1-K_MC15TuneV1_ProbNNp), B_M0123_Subst01_Kpi2pK. The overtraining check of the results is shown in Figure4.35 and the ROC curve is shown in Figure4.36.

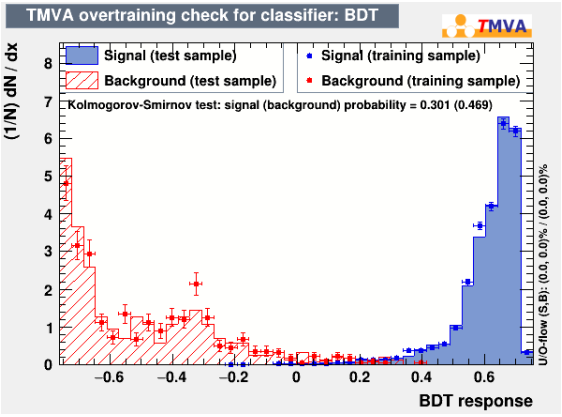

Figure 4.35: The overtraining check for classifier.

From the figure above it can be seen that the training is not so good. When

background efficiency is 0.01, the signal efficiencies from test and training sample are 0.982 and 0.988; when background efficiency is 0.10, the signal efficiencies from test and training sample are 1.000 and 1.000; when background efficiency is 0.30, the signal efficiencies from test and training sample are 1.000 and 1.000.

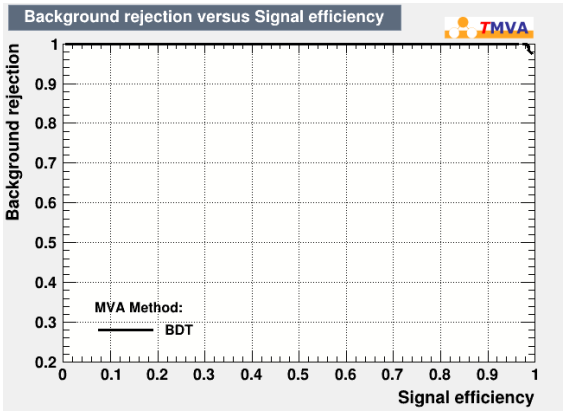

Figure 4.36: The ROC curve of BDT method.

The area integral under the ROC curve is 1.000.

4.3 Compare the BDT method with the cut-based method

For the K-e misidentification background in the J/ψ e-h swap background, using a cut-based method, the signal efficiency of the best cut is 0.737, and the background rejection of the best cut is 0.978. The comparison between the cut-based method and the BDT method(using the second set of input variables) is shown in Figure4.37.

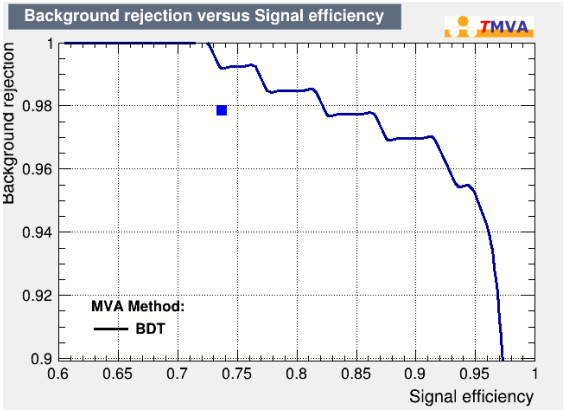

Figure 4.37: Curve: the ROC curve from the BDT method. Blue dot: the result from the cut-based method.

Using a BDT method, when the signal efficiency is 0.737, the background rejection is improved to 0.992, and when the background rejection is 0.978, the signal efficiency is improved to 0.820.

For the π-e misidentification background in the J/ψ e-h swap background, using a cut-based method, the signal efficiency of the best cut is 0.735, and the background rejection of the best cut is 0.949. The comparison between the cut-based method and the BDT method(using the second set of input variables) is shown in Figure4.38.

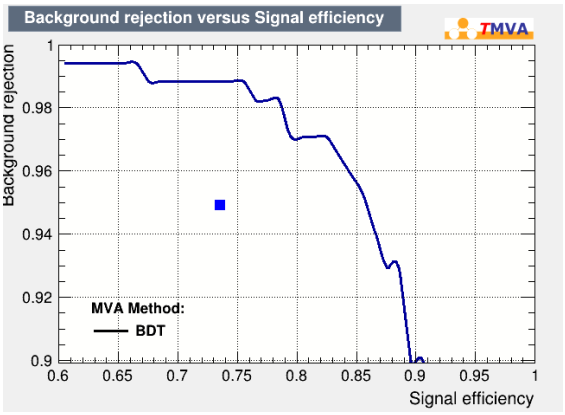

Figure 4.38: Curve: the ROC curve from the BDT method. Blue dot: the result from the cut-based method.

Using a BDT method, when the signal efficiency is 0.735, the background rejection is improved to 0.989, and when the background rejection is 0.949, the signal efficiency

is improved to 0.860.

For the $\pi \to p$ misidentification background in the $\Lambda_b^0 \to pK^- J/\psi(\to e^+e^-)$ background, using a cut-based method, the signal efficiency of the best cut is 0.734, and the background rejection of the best cut is 0.981. The comparison between the cut-based method and the BDT method(using the second set of input variables) is shown in Figure4.39.

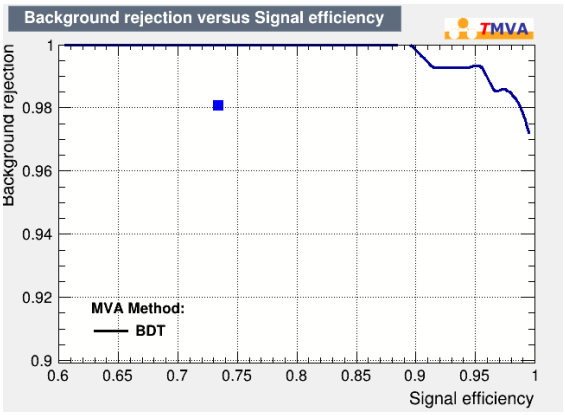

Figure 4.39: Curve: the ROC curve from the BDT method. Blue dot: the result from the cut-based method.

Using a BDT method, when the signal efficiency is 0.734, the background rejection is improved to 1.000, and when the background rejection is 0.981, the signal efficiency is improved to 0.980.

For the $K\pi \to pK$ misidentification background in the $\Lambda_b^0 \to pK^- J/\psi(\to e^+e^-)$ background, using a cut-based method, the signal efficiency of the best cut is 0.736, and the background rejection of the best cut is 0.983. The comparison between the cut-based method and the BDT method(using the second set of input variables) is shown in Figure4.40.

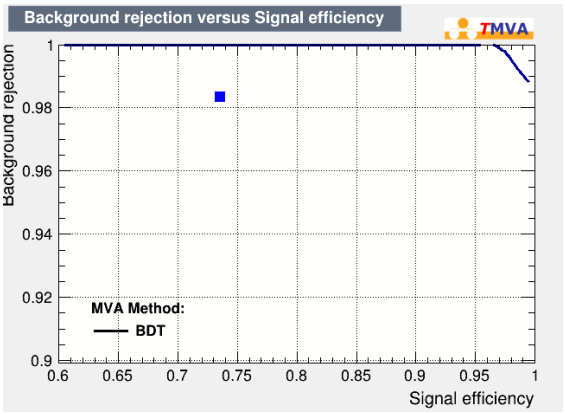

Figure 4.40: Curve: the ROC curve from the BDT method. Blue dot: the result from the cut-based method.

Using a BDT method, when the signal efficiency is 0.736, the background rejection is improved to 1.000, and when the background rejection is 0.983, the signal efficiency is improved to 1.000.

Chapter 5

Conclusion and Comment

In this analysis, the cut-based method and the BDT method are used to study the selection of $B^0 \to K^+\pi^-e^+e^-$. Two kinds of backgrounds from $B^0 \to K^*(\to K^+\pi^-)J/\psi(\to e^+e^-)$ and two kinds of backgrounds from $\Lambda_b^0 \to pK^-J/\psi(\to e^+e^-)$ are discussed.

For the K-e misidentification background in J/ψ e-h swap background, using a cut-based method, the best cut is (4580 < Jpsi_cons_mix_cov_double_kemisid_B_M < 4601) & (L1_MC15TuneV1_ProbNNe < 0.21), and the signal efficiency and the background rejection are 0.737 and 0.978. Using a BDT method, the background rejection is improved to 0.992 under the same signal efficiency, and the signal efficiency is improved to 0.820 under the same background rejection.

For the π-e misidentification background in J/ψ e-h swap background, using a cut-based method, the best cut is (3500 < Jpsi_cons_mix_cov_double_piemisid_B_M < 4400.6) & (L2_MC15TuneV1_ProbNNe < 0.288), and the signal efficiency and the background rejection are 0.735 and 0.949. Using a BDT method, the background rejection is improved to 0.989 under the same signal efficiency, and the signal efficiency is improved to 0.860 under the same background rejection.

For the $\pi \to p$ misidentification background in $\Lambda_b^0 \to pK^-J/\psi(\to e^+e^-)$ background, using a cut-based method, the best cut is (4000 < B_M0123_Subst1_pi2p < 5100.6) & (Pi_PIDp>0), and the signal efficiency and the background rejection are 0.734 and 0.981. Using a BDT method, the background rejection is improved to 1.000 under the same signal efficiency, and the signal efficiency is improved to 0.980 under the same background rejection.

For the $K\pi \to pK$ misidentification background in $\Lambda_b^0 \to pK^-J/\psi(\to e^+e^-)$ background, using a cut-based method, the best cut is (4000 < B_M0123_Subst01_Kpi2pK < 5025.5) & (Pi_PIDK>5.6), and the signal efficiency and the background rejection

are 0.736 and 0.983. Using a BDT method, the background rejection is improved to 1.000 under the same signal efficiency, and the signal efficiency is improved to 1.000 under the same background rejection.

From the results above it can be seen that the BDT method is better than the cut-based method on the selection of these four kinds of backgrounds, and the background rejections and the signal efficiencies are improved using the BDT method.

Bibliography

[1] The LHCb Collaboration. The LHCb detector at the LHC. *Journal of Instrumentation*, 3(08):S08005–S08005, aug 2008.

[2] LHCb Tracker Upgrade Technical Design Report. Internal documentation, 2014.

[3] Lhcb detector performance. *International Journal of Modern Physics A*, 30(07):1530022, Mar 2015.

[4] Barbara Storaci. Optimization of the LHCb track reconstruction. *Journal of Physics: Conference Series*, 664(7):072047, dec 2015.

[5] S. L. Glashow, J. Iliopoulos, and L. Maiani. Weak interactions with lepton-hadron symmetry. *Phys. Rev. D*, 2:1285–1292, Oct 1970.

[6] N. Cabibbo. Unitary symmetry and leptonic decays. *Physical Review Letters, 10, 531.*, 1963.

[7] Makoto Kobayashi and Toshihide Maskawa. CP-Violation in the Renormalizable Theory of Weak Interaction. *Progress of Theoretical Physics*, 49(2):652–657, 02 1973.

[8] Andrzej J. Buras. Flavour Changing Neutral Current Processes. *arXiv e-prints*, pages hep–ph/9610461, October 1996.

[9] A. Ali. Flavour changing neutral current processes in b decays. *Nuclear Physics B - Proceedings Supplements*, 59(1):86–100, 1997. Proceedings of the Fourth KEK Topical Confernce on Flavor Physics.

[10] Bettler M.-O. Owen P. Petridis K. A. Archilli, F. Flavour-changing neutral currents making and breaking the standard model. *Nature*, 2017.

[11] Wolfgang Altmannshofer. New physics in $B \to K^* \mu\mu$. *The European Physical Journal C*, 2013.

[12] Gudrun Hiller and Frank Krüger. More model-independent analysis of$b \to s$processes. *Physical Review D*, 69(7), Apr 2004.

[13] Christoph Bobeth, Gudrun Hiller, and Giorgi Piranishvili. Angular distributions of $\bar{B} \to \bar{K}\bar{l}l$ decays. *Journal of High Energy Physics*, 2007(12):040–040, Dec 2007.

[14] Chris Bouchard, G. Peter Lepage, Christopher Monahan, Heechang Na, and Junko Shigemitsu. Standard model predictions for $B \to Kl^+l^-$ with form factors from lattice qcd. *Physical Review Letters*, 111(16), Oct 2013.

[15] Gudrun Hiller and Martin Schmaltz. Diagnosing lepton-nonuniversality in $b \to sll$. *Journal of High Energy Physics*, 2015(2), Feb 2015.

[16] R. Aaij, B. Adeva, M. Adinolfi, Z. Ajaltouni, S. Akar, J. Albrecht, F. Alessio, M. Alexander, S. Ali, and et al. Test of lepton universality with $B^0 \to K^{*0}l^+l^-$ decays. *Journal of High Energy Physics*, 2017(8), Aug 2017.

[17] Rutgers University Tong Zhang. Decision tree and boosting.

https://cilvr.cs.nyu.edu/diglib/lsml/lecture03-trees-boosting.pdf.

[18] Yann Coadou. Boosted decision trees.

https://indico.cern.ch/event/472305/contributions/1982360/attachments/1224979/1792797/ESIPAP_MVA160208-BDT.pdf, 2016.

[19] Katherine Woodruff. Introduction to boosted decision trees.

https://indico.fnal.gov/event/15356/contributions/31377/attachments/19671/24560/DecisionTrees.pdf, 2017.

[20] Gaurav. An introduction to gradient boosting decision trees.

https://www.machinelearningplus.com/uncategorized/an-introduction-to-gradient-boosting-decision-trees/, 2021.

[21] Rie Johnson and Tong Zhang. Learning nonlinear functions using regularized greedy forest, 2014.

[22] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28:337–407, 04 2000.

[23] Jerome Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29, 11 2000.

[24] Jerome Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38:367–378, 02 2002.

[25] Jerome Friedman Trevor Hastie, Robert Tibshirani. The elements of statistical learning. *Springer*, 2009.

[26] Andreas Hoecker, Peter Speckmayer, Joerg Stelzer, Jan Therhaag, Eckhard von Toerne, and Helge Voss. TMVA: Toolkit for Multivariate Data Analysis. *PoS*, ACAT:040, 2007.

[27] R. Aaij, B. Adeva, M. Adinolfi, Z. Ajaltouni, S. Akar, J. Albrecht, F. Alessio, M. Alexander, S. Ali, G. Alkhazov, and et al. Measurement of the b-quark production cross section in 7 and 13 tev pp collisions. *Physical Review Letters*, 118(5), Feb 2017.

[28] Y. Amhis, Sw. Banerjee, E. Ben-Haim, F. U. Bernlochner, M. Bona, A. Bozek, C. Bozzi, J. Brodzicka, M. Chrzaszcz, J. Dingfelder, and et al. Averages of b-hadron, c-hadron, and τ-lepton properties as of 2018. *The European Physical Journal C*, 81(3), Mar 2021.

[29] P.A. Zyla et al. Review of Particle Physics. *PTEP*, 2020(8):083C01, 2020.

[30] R. Aaij, B. Adeva, M. Adinolfi, A. Affolder, Z. Ajaltouni, J. Albrecht, F. Alessio, M. Alexander, S. Ali, and et al. Study of the kinematic dependences of Λ_b^0 production in pp collisions and a measurement of the $\Lambda_b^0 \to \Lambda_c + \pi^-$ branching fraction. *Journal of High Energy Physics*, 2014(8), Aug 2014.