RICE UNIVERSITY

A Comparison of Multivariate Data Analysis Techniques as Applied to the Identification of Electrons and Tau Leptons

by

Andrew W. Askew

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

MASTER OF SCIENCE

Approved, Thesis Committee:

B. Paul Padley, Chair
Faculty Fellow of Physics and Astronomy

Hannu E. Miettinen
Professor of Physics and Astronomy

Richard Wolf
Professor of Physics and Astronomy

Houston, Texas

May, 2001

ABSTRACT

A Comparison of Multivariate Data Analysis Techniques as Applied to the Identification of Electrons and Tau Leptons

by

Andrew W. Askew

This thesis compares the performance of Probability Density Estimation and Neural Networks as applied to the identification of tau leptons and electrons at the DØ detector for Run II. The theory behind each method of multivariate analysis is briefly described. The efficiencies of each of the methods are compared from analysis of Monte Carlo data samples, and optimal choices for the discrimination between signal and background are made.

Acknowledgments

I would like to thank my advisor, B. Paul Padley for his patience in going through the numerous iterations of thesis writing, as well as for the impetus to go ahead and write on this topic. I would also like to thank Hannu Miettinen for some valuable insights into the PDE process, and Bryan Smith for help on the optimization routines.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Overview of the Standard Model

The Standard Model of Particle physics is a detailed phenomenological description of the interaction between the fundamental particles of nature. Its predictions agree with all experiments to date.

The Standard Model states that there are only two types of particles in nature, fermions and bosons. Fermions have half-integer spins, and make up all the matter in the universe. Bosons have integer spins, and are assigned the task of mediating the fundamental forces of nature, through their emission and absorption by fermions. The Standard Model describes only the interaction of the strong, weak, and electromagnetic forces, gravity is not included.

Fermions are divided into two families, leptons and quarks, which are each divided into three generations. Tau leptons, muons, and electrons only interact through the electromagnetic and weak forces, and are paired up with their corresponding neutrinos which interact through the weak force. Table 1.1 shows the three lepton families and their properties.

There are three generations of quarks which participate in all of the Standard Model forces, and are shown in Table 1.2 with their masses and charges. Quarks possess an additional internal degree of freedom called color. Color charge affects

Lepton	Mass [GeV/c^2]	Charge [e]	Lifetime
e	0.5110	-1	$>4.2\times10^{24}yr$
ν_e	$< 15\cdot10^{-6}$	0	
μ	105.7	-1	$2.2\times10^{-6}s$
ν_μ	< 0.17	0	
τ	1777	-1	$2.9\times10^{-13}s$
ν_τ	< 24	0	

Table 1.1 Fundamental fermions in the Standard Model: leptons.

how quarks interact with one another through the strong force, which is described by Quantum Chromodynamics (QCD). To date, no particle with a net color has been observed, due to the fact that the strong force is non-Abelian (i.e. the gluons mediating the strong force interact with themselves as well as with the quarks). The resulting increase of strength with distance makes it energetically favorable at increasing inter-quark distances to allow the creation of quark-antiquark pairs, which hadronize the two quarks and conceal the color charge. The production of jets of particles through QCD processes accounts for a large number of the interactions that occur at collider experiments. This is because the cross-section for QCD processes increases with the number of possible final states, and therefore the energy. The many different shapes and energies of jets from QCD processes increase the difficulty of identifying phenomena with smaller cross-sections. These jets provide difficult backgrounds to discriminate against when searching for new physics.

The integer spin bosons are the mediators of Standard Model forces. The fermions

Quark	Mass [GeV/c^2]	Charge [e]
u	2–8 $\cdot 10^{-3}$	2/3
d	5–15 $\cdot 10^{-3}$	-1/3
c	1.0–1.6	2/3
s	0.1–0.3	-1/3
t	180	2/3
b	4.1–4.5	-1/3

Table 1.2 Fundamental fermions in the Standard Model: quarks.

interact with each other through exchange of these particles. Quantum Electrodynamics (QED) is an example of a description of a force in terms of a gauge theory (electromagnetic interaction through exchange of virtual photons). The weak interaction is a gauge theory with one important difference: The mediating particles for the weak interaction possess non-zero, large masses. The mass of the W and Z bosons are given by their interaction with the Higgs boson.

Gauge boson	Mass [GeV/c^2]	Charge [e]
gluons	0	0
γ	0	0
$W^{\pm}$	80.45	± 1
Z^0	90.91	0

Table 1.3 Gauge bosons in the Standard Model.

Interaction with the Higgs boson is proposed to be the method by which all of the Standard Model particles gain their masses. Mass limits have been set on the Higgs in collider experiments, but the last undiscovered particle in the Standard Model remains undetected. The identification of the Higgs, along with the measurement of its mass will be an important test of the Standard Model.

1.2 Motivation for Identification of Electrons and Tau Leptons

Run II at Fermilab will be the first opportunity for collider experiments to take advantage of upgrades to the Tevatron. These upgrades will allow for an order of magnitude increase in luminosity over the previous run. Also, the beam energy for Run II will increase to 2 TeV, over the Run I 1.8 TeV. The increase in luminosity and energy will provide an opportunity for physicists to study particle collisions at the highest energy in the world.

To study the states that are produced, and interactions that occur at these high energies, it is necessary to correctly identify the final particle states in the detector. Electrons have been found with high efficiency in the past, and greater efficiency will improve the number of channels that may be examined for new processes. Electrons are found in final states of many processes. The identification of tau leptons was too difficult in the past to be fully exploited, and as a result, some channels for new physics that have not been explored depend heavily on decays involving tau leptons. Two are briefly described below.

Run II at the Tevatron will be an opportunity to explore new parameter space to identify the only unobserved Standard Model boson. The coupling of the Higgs to the other known particles is proposed to be the mechanism by which each particle acquires mass. Once the Higgs mass is measured, the Standard Model predicts all of the branching ratios of the Higgs to all other particles. For allowed Higgs masses that are observable at DØ, the predominant decay of the Higgs is H→bb. However, with all of the branching ratios of the Higgs predicted by its mass, the measurement of the decay of the Higgs to the next heaviest Standard Model particles, c quarks and tau leptons, will provide an important check on the identification of the Higgs. Thus the identification of tau leptons will play an important role in the detection of the Higgs.

The search for new phenomena beyond the Standard Model will also be under way. The most popular of these proposed new phenomena is Supersymmetry (SUSY). Supersymmetry is a symmetry between fermions and bosons. The concept is that all Standard Model fermions have spin-1 partners, and that all the bosons have spin-1/2 partners, eliminating the asymmetry in the number of fermions and bosons. With the presence of SUSY particles in Standard Model processes, some difficulties in higher energy regimes (infinities that must be renormalized away) are eliminated. The detection of these supersymmetric partners would signal new physics beyond the current Standard Model and open up a new range of processes to study. SUSY signals generally decay to the lightest supersymmetric particle (LSP) which does not interact with the detector and escapes. The branching fractions of the chargino (the supersymmetric partner of the charged Higgs boson), in the remaining parameter space has a large number of leptonic channels, most prominently to three tau leptons. Therefore the identification of taus will play an important part in the search for supersymmetry.

1.3 Tau Lepton and Electron Identification

1.3.1 Experimental Signitures of Leptons

Muons are identical to electrons, except for their mass. The large mass of the muon causes the threshold of bremstrahlung radiation in matter to be much higher, and as a result muons do not cause electromagnetic showers. Muons are identified as the particles that pass through the entire detector with a minimum of energy deposited. Generally the detection and measurement of the muons are made in the outermost regions of the detector. Magnetized iron, along with tracking equipment such as scintillators or drift chambers, gives a measurement of the curvature of the muon's trajectory, due to the applied magnetic field, and thus a measurement of its momentum as it leaves the detector.

Electrons are not as simple to detect. They can be mistaken for photons, since the shower formed from electromagnetic pair production in material has the same characteristics for electrons and photons. Tracking information simplifies this task, since photons leave no track, but other processes (such as QCD jets) can still resemble electrons. A single jet containing a charged particle which provides a track as well as an uncharged pion, which decays into two photons providing an electromagnetic shower, can mimic an electron.

Tau leptons are difficult to identify. Unlike electrons which are stable, and muons which have lifetimes long enough to escape the detector, tau leptons decay quickly ($\tau_\tau = 2.9 \times 10^{-13}s$). The majority of their decays ($\approx$ 65%) contain hadrons which may be confused with particle jets from QCD processes. Selected tau decay modes are presented in Table 1.4. The proper identification of the other two generations of leptons is important to reconstruct the leptonic decays of taus.

τ decay	Fraction($\frac{\Gamma_i}{\Gamma}$)
$\tau^- \to e^- \overline{\nu_e} \nu_\tau$	17.8%
$\tau^- \to \mu^- \overline{\nu_\mu} \nu_\tau$	17.3%
$\tau^- \to \pi^- \nu_\tau$	11.1%
$\tau^- \to \pi^- \pi^0 \nu_\tau$	25.4%
$\tau^- \to \pi^- \pi^+ \pi^- \nu_\tau$	9.49%

Table 1.4 Selected τ decay modes.

Since neutrinos only participate through the weak interaction, their interaction cross section is very small, and thus no direct signals are found in the detector. The neutrinos from particle decays are detectable only through the missing energy in the detector.

1.3.2 Identification techniques

Using either real data from known processes, or simulated data based on the detector's reaction to chosen processes, the efficiency of correctly identifying particles may be studied. Traditionally this separation has been made by placing linear cuts on selected quantities that were chosen to be characteristic of the given particle (quantities that show a good separation between the process being studied and processes that overall are similar). This has the advantage of being a visual method, one may form histograms of the data and observe how many signal events are accepted by the cut and how many background events are rejected.

Multivariate techniques such as the covariance comparison (Hmatrix) have also been used with some success. In this method, a discriminant is found by comparing the inverse of the covariance matrices of the signal and background. In the previous detector run, this method was used for the identification of electrons and tau leptons. Neural networks have been used in the past, with good efficiency, in the selection of signal events. Early results have shown that neural networks give a greater efficiency for particle identification than that of Hmatrix [1].

Probability density estimation and neural networks, while less visual than linear cuts and more complex than covariance comparisons, can better handle the correlations that arise between the different quantities measured. These processes treat each event as a point within a variable space of the number of quantities considered, rather than a single dimension at a time when linear cuts are made. The advantage of this is that the events are selected on the basis of how well they match the desired signal as a whole rather than if they are within the confines of a linearly defined space.

Neural networks depend on a large number of free parameters that are determined by an iterative algorithm to produce a desired output for a given signal. Probability density estimation develops functions of the input parameters that are characteristic of the signal and background. Data events are then selected by using a discriminant formed from these feature functions.

Chapter 2
Detector Description

An overview of the upgraded DØ detector is presented. Interested readers are referred to [2] for a detailed description of the Run I detector and [3] for the enhancements added for Run II. The major components are described with an emphasis on the instrumentation that will be used in the measurement of quantities used in the offline multivariate analysis.

Note that two different coordinate systems are used in the description of the detector. The first is a conventional cylindrical-polar description in which the z-axis is identified with the beam line through the center of the detector. The radial direction and the ϕ-angle are then used for identifying the position in the detector as a whole. There is another system used however, coordinates in η and ϕ are quoted where ϕ is simply the azimuthal coordinate and η is called the pseudorapidity and is defined as;

$$\eta = -ln(tan(\frac{\theta}{2})), \tag{2.1}$$

where θ is the polar angle made with respect to the beam axis. In the high energy limit (p$\gg$m), η is a good approximation of the true rapidity (y) which is given by:

$$y = \frac{1}{2} ln \frac{E + p_z}{E - p_z}. \tag{2.2}$$

The advantage of using the rapidity (or in our case the pseudorapidity) is that distribution of particles is approximately flat in η.

2.1 Tracking

The Run I DØ detector consisted of the calorimeter, the muon system and a central transition radiation detector. There was no magnetic field in the interaction region to provide charge determination or a momentum measurement. The transition radiation detector has been removed for Run II and replaced by the silicon vertex detector, the central fiber tracker and a superconducting solenoid. The silicon vertex detector will provide good spatial resolution in the area very close to the beam line. The central fiber tracker will provide a fast trigger for tracks in the detector as well as a measurement of momentum (curvature) of the tracks in the magnetic field.

2.1.1 Silicon Vertex Detector

The silicon vertex detector (SVD) is the closest subsystem to the beam pipe. Its placement allows for coverage of $|\eta| < 3$. The purpose of the SVD is to allow tracking very close to the beam line so that the vertex of the event taking place may be reconstructed. The secondary vertices from subsequent decay of short lived particles may also be reconstructed.

The SVD is made up of a number of modules that utilize small strips of silicon with fine cathode lines etched into their surfaces. These silicon microstrips are solid state detector devices in which electron-hole pairs are produced due to the passage of a charged particle. When the electrons are collected at the nearest cathode, a signal is produced which will determine the place at which the particle passed through the silicon. Reconstruction of the path of the particle leaving the interaction region is made based on these signals.

To cover this range in η the detector was broken up into different modules placed along the beamline: small diameter barrels, slightly larger diameter "F" disks, and large "H" disks. The four barrels closest to the center of the detector have four layers of double sided silicon microstrips in the axial direction and at ninety degrees to the axial direction (this direction indicates the direction of the cathodes). The outermost two barrels have single sided silicon strips with strips in the axial direction and two degrees to the axial. The "F" disks are made up of double sided silicon strips at thirty degrees stereo pitch. The "H" disks are large diameter and are made up of single sided silicon strips at fifteen degrees stereo. The outermost part of the silicon vertex detector are the "H" disks, with two on each end. Then four "F" disks are placed between the "H" disks and the outermost barrels. The remaining "F" disks are interspersed between the barrels in the central region. The silicon vertex detector is expected to allow for spatial resolution of approximately 10 μm.

2.1.2 Central Fiber Tracker

The Central Fiber Tracker (CFT) is made up of concentric cylinders of scintillating fibers, which are located just outside the SVD and just inside the solenoid. Photons are produced in the scintillating fiber which are then transmitted along the fiber which also acts as a waveguide. The ends of the fibers are coupled to optical fibers that carry the light to photon counters.

The CFT is made up of eight superlayers of multiclad scintillating fibers. Each superlayer is made up of a layer of axial fibers and a layer of stereo fibers at two degrees pitch. A charged particle passing through the fiber layers deposits energy through ionization in the fiber. A small percentage of this energy is radiated from the organic molecules (through radiative decay back to their ground states, just like in a conventional scintillator) as visible light photons. These photons travel down to the end of the fiber where the light passes into a waveguide and is carried into a specially prepared cassette.

These cassettes contain arrays of visible light photon counters (VLPCs) which convert the photon signals to electrical signals much like a photomultiplier tube (with gains on order 20,000-50,000 and high quantum efficiency). The operating temperature of these devices is approximately 7 K. Therefore a supply of liquid helium is required for the operation of the cassettes. Resistive heaters and temperature sensors have been placed near the VLPCs to ensure that the optimum temperature is maintained. The design of the cassettes is quite complex, due to the necessity of keeping the portion of the device with the VLPCs at low temperature, while carrying the signals from the chips to the readout electronics at room temperature. The interested reader is referred to [4] for further details.

The CFT will provide a measurement of the momentum of the charged particles passing through it, via the curvature of the tracks from the uniform magnetic field. This subsystem will also provide a low level (L1) hardware trigger using just the axial layers of the fibers to reconstruct tracks via logic preloaded into field programmable gate arrays (FPGAs). These FPGAs are static RAM driven, and thus the momentum thresholds may be set online. The position resolution of the CFT is approximately 100 μm. The expected momentum resolution of the CFT has been parametrized:

$$\frac{\sigma_{p_T}}{p_T} = \sqrt{0.015^2 + (0.0014 p_T^2)}. \tag{2.3}$$

2.1.3 Solenoid

The inclusion of a magnetic field increases the number of processes that may be explored by providing charge discrimination. The solenoid surrounds the new tracking subsystems, and is located just interior to the preshower detectors and calorimeter.

The superconducting solenoid magnet will keep a constant field (to approximately 0.5%) of 2 Tesla in the region of the tracking subsystems. It is a conventional superconducting solenoid, which requires a supply of liquid helium to keep the temperature low enough for the coil to conduct without resistance. Since there is no return path for the field lines, the field is kept constant in the end regions by using different grades of conductor, allowing for a greater charge density at the ends of the solenoid than in the center. The liquid helium supply that is provided to the magnet will also supply the cryostats necessary for the fiber tracker/preshower detector cassettes (described in the fiber tracker section). The energy stored in the solenoid is approximately 5 MJ. The bulk of the coil and cryostat that make up the coil assembly is approximately 0.87 radiation lengths thick.

2.2 Energy Measurement

The sampling calorimeter from Run I is unchanged, except for the upgrade in the readout electronics which allows the calorimeter to operate in the new higher luminosity and higher crossing time of Run II. Central and forward preshower detectors have been added to provide additional energy sampling and shower profiles prior to the calorimeter.

2.2.1 Preshower Detector

The preshower system is separated into two parts, the central preshower (CPS) and the forward preshower (FPS). The function of both the CPS and FPS is to provide shower sampling (and therefore energy sampling) just prior to the calorimeter, as well as providing additional tracking information outside the solenoid.

The preshower detectors are made up of triangular strips of scintillating material that are wrapped about wavelength shifting fiber optic cable. An electron or photon entering the preshower detector has already begun to shower, since passing through the inner tracking systems. Photons from the scintillating material enter the fiber optic cable, where by absorption and re-emission at smaller wavelengths, the photons are shifted into the optimum range to be read out by VLPCs in cassettes like those described in the CFT. Here, there is expected to be a considerably larger light yield than in the CFT system, and thus the VLPCs do not require as high an efficiency. Using these strips the showers that began in the inner subsystems can be sampled prior to the calorimeter. Muons will not shower, but will still leave a track in the preshower detector.

The division of the calorimeter sections necessitates the division of the preshower detector into two regions, one prior to the central calorimeter, the others located on the endcaps. A tapered layer of lead is placed just prior to the scintillator strips so the amount of material that the particles pass through prior to reaching the preshower system is equalized. In addition to the two preshower systems, a layer of scintillating pixels was added in the remaining space between the solenoid cryostat and the calorimeter cryostat so that there would be no inactive medium for the particles to pass through. This gives additional position measurements for all charged particles just prior to the calorimeter.

2.2.2 Calorimeter

The DØ calorimeter is a liquid argon-uranium sampling compensating calorimeter that has been well studied since DØ first went online. The calorimeter is formed by the central calorimeter, which covers the range $|\eta| < 1.1$ and the two endcap calorimeters which cover out to $|\eta| \approx 4$. Liquid argon is the active medium here, particles passing through the absorber material (the uranium, and in some areas copper and steel) will shower. Then the subsequent shower particles ionize the liquid argon providing a signal through the high voltage readout pads located on the absorber plates.

The calorimeter is divided into three different sections of segmentation. There are the four the innermost layers, which make up the electromagnetic calorimeter. These layers are finely segmented, and the absorber is almost pure uranium. Next, there are three layers of larger segmentation and thicker uranium-niobium alloy plates which make up the fine hadronic portion of the calorimeter. Beyond the fine hadronic section there is a single layer of coarsely segmented absorber made up of thick copper (and stainless steel in the case of the endcaps) plates which serve as the coarse hadronic layer.

The electromagnetic calorimeter layers are designed so that showers from electrons and photons stop within the finely segmented layers of uranium. The maximum of electromagnetic showers is calculated to take place in the third electromagnetic layer, so this region is further segmented for a more precise measurement.

The DØ calorimeter is a compensating calorimeter. The principle of a compensating calorimeter is to equalize the response of the calorimeter signals from hadrons and electrons. This is achieved by the tuning the thickness and the material chosen for the absorber. The depleted uranium absorber provides backgrounds at low energies due to its own radioactive breakdown, but low energy neutrons from nuclear breakup (due to the hadronic showers) cause fission in the uranium producing charged particles which compensate for the lower visible energy in hadronic showers. The ionization energy from these fission effects can be measured in the same way as the ionization energy from leptons.

The calorimeter is organized into towers for readout. Each 'tower' in the calorimeter is approximately 0.1×0.1 in $\eta \times \phi$, and can thus be used to get some measure of shower or jet profile while online. The thresholds of the energy deposited in the towers of the calorimeter are set as part of the triggering of the data aquisition of the detector. The resolution of the calorimeter can be parametrized as;

$$(\frac{\sigma_E}{E})^2 = C^2 + \frac{S^2}{E} + \frac{N^2}{E^2}, \tag{2.4}$$

where for electrons C=0.003, S=0.157, and for pions C=0.032 and S=.41, and N is the noise contribution from the electronics and the background radiation from the uranium. For hadron showers, it is more difficult to gain an accurate measurement of the energy deposited, since the showering mechanism is different (strong force interactions rather than electromagnetic).

The only change from the calorimeter from Run I was a replacement of the readout electronics which allows for faster readout. This replacement was necessary due to the increased luminosity and crossing time of Run II.

2.3 Muon System

The DØ muon system has been upgraded since the first detector run. Here a brief summary of the components of the Run II muon system is presented.

The muon system is divided into two sections, the wide angle section ($|\eta| < 1$) and the forward section ($1 < |\eta| < 2$). Each of these systems is divided into three layers, A, B, and C, with an iron toroidal magnet between layers A and B. The central (or wide angle) muon system consists of three layers of proportional drift tubes, with scintillators to provide the necessary time stamp for the chambers. This allows for momentum measurement and tracking independent of the central systems. The forward muon system is composed of mini-drift tubes instead of proportional drift tubes due to the higher radiation environment in the forward direction. From Run I, it was discovered that the proportional drift chambers had to be 'cleaned' by the application of a high voltage to return to the cathodes to their proper operating conditions. Due to the high occupancy in the forward direction it was necessary to 'clean' these chambers with an unacceptable frequency (allowing too much dead time) and thus the replacement with the mini-drift tubes. The only difference between these tubes and the larger proportional drift chambers will be the lower occupancy due to the much smaller cross-sectional area of the small tubes, as well as an increase in the response time. For both of the muon systems the resolution is on order 1mm due to multiple scattering.

Chapter 3
Multivariate Techniques

3.1 Neural Networks

Neural networks are effective in event selection. When trained on signal and background data, the output can be used to choose the best events to use for analysis. A brief discussion of neural networks, their basic mathematics and in particular the methods used in the analysis for particle identification is presented. For more information, the reader is referred to [5].

3.1.1 Mathematics

The neural network most often used for pattern recognition is called a multi-layer perceptron. This structure is made up of a number of inputs, a layer (or multiple layers) of hidden nodes, and one or more outputs. Each input is connected to each node in the hidden layer by a weight, and then each node in the hidden layer computes an output based on a sigmoid function of the linear combination of inputs. Each hidden node is connected to each output node by another weight, and thus an output is computed at each output node either by a linear combination of the output from each hidden node multiplied by another weight, or a sigmoid function of that linear combination.

For a linear combination of input variables, a single layer of sigmoidal hidden nodes

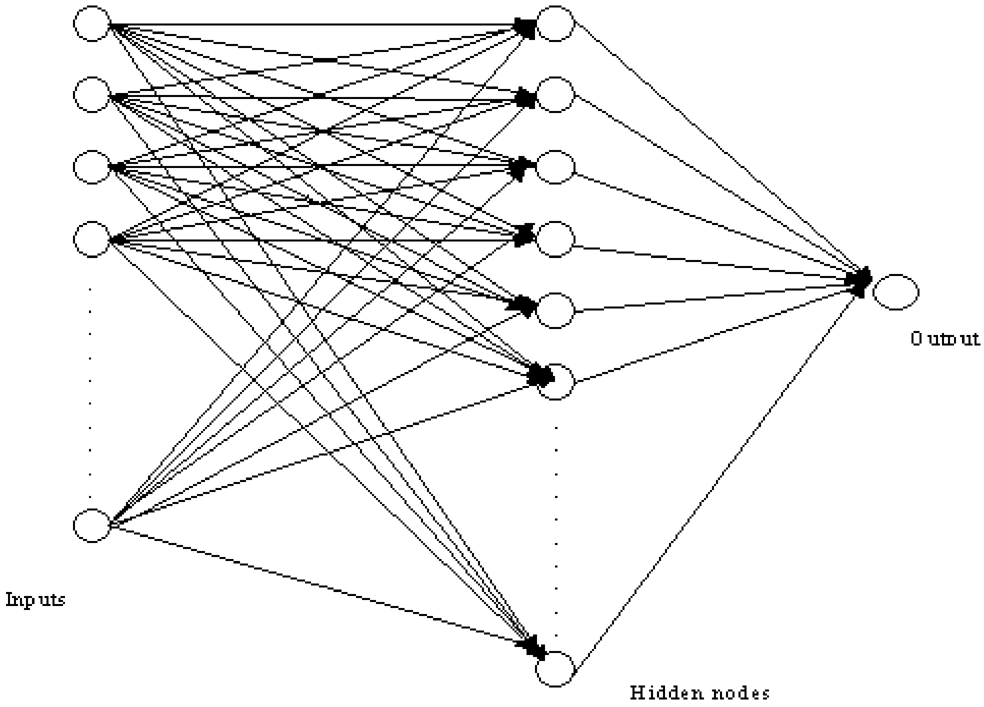

Figure 3.1 Pictorial Structure of a Neural Network

and a single output composed of a linear combination of the hidden layer outputs, the output of the neural network is;

$$y = \sum_{j=1}^{N_{nodes}} (g(\sum_{i=0}^{N_{inputs}} \omega_{ij} x_i) \times \omega_j) + O, \tag{3.1}$$

where y is the output, O is the output offset, ω_{ij} is the weight of the ith input to the jth hidden node, ω_j is the weight of the jth hidden node to the output, and g(.) is the activation function of the hidden layer nodes, usually a sigmoid function (a function that varies between one and zero) such as:

$$g(x) = \frac{1}{1 + e^{-x}}. \tag{3.2}$$

The principle behind neural networks is that any function of n variables can be mapped into a defined pattern on the output by a linear combination of sigmoid functions with appropriate coefficients. The output of the above mathematics then is a linear combination of sigmoid functions (from the hidden nodes) as a function of the input nodes, with coefficients given by the weights. The error of a neural network is defined:

$$E_j = \sum_i (y_i - t_j)^2. \tag{3.3}$$

Here, E_j is the error of the jth training pattern, y_i is the output from the neural network for the ith vector of inputs in the training sample for pattern j, and t_j is the desired output from the neural network for pattern j. Therefore, for the case of separating two patterns, signal and background, two sets of data are required. The error function for the given weights can then be computed, and adjustments made in the free parameters so that the optimum performance is obtained. For discrimination between signal and background, neural networks are generally chosen to have a value of one on the output for signal and a value of zero on the output for background. In this way, the ideal case is to form a distribution for subsequent linearly independent sets of testing data using the trained neural network, that peaks about zero and one. Then a single cut can be placed on this output that will separate the signal from the background.

3.1.2 Implementation

The neural network analysis was performed using the Physics Analysis Workstation (PAW) package, version 2.11/11, which includes the MLPfit neural network suite version 1.33. For more details on the mathematics of the method by which the weights of the neural network were optimized, the reader is referred to Appendix A. The parameters that need to be optimized for each individual analysis are the number of nodes used in the network, and the number of cycles that the network was trained over.

3.2 Probability Density Estimation

The Probability Density Estimation (PDE) method was developed at Rice University for the top quark search [7]. For that analysis, its efficiency was found to be comparable to that of neural networks. A brief discussion of the PDE method, as well as the new adaptive kernel, a modification of the original algorithm, follows.

3.2.1 Mathematics

The standard (fixed kernel) probability density estimation method of multivariate data analysis has been previously documented. This method is much different than the previously mentioned case of neural networks as it has few free parameters, allowing for less complicated optimization. Given a training sample of data, consisting of a set of signal and a set of background, two functions of the n input variables are formed of the data set. This is accomplished by forming a product of kernel functions in the space of the input variables for each data point. The complete function for the entire sample of events is the sum of all of these product kernels in the training data, normalized by the number of training events used to form the function. This is known as the feature function, because it is an estimate of the important features in the data. Mathematically this function is given by:

$$f(\mathbf{x}) = \frac{1}{N_{tr} h_1 \ldots h_d} \sum_{i=1}^{N_{tr}} [\prod_{j=1}^{d} K(\frac{x_i - x_{ij}}{h_j})]. \tag{3.4}$$

The $\mathbf{x}$ are the set of variables this analysis is being performed on. These are the initial input variables chosen for the analysis after transformation by linear algebra methods into a set in which the first order correlations vanish (the new set of variables is a linear combination of the original variables). This transformation is done so that the kernel structure will match the covariance structure of the data, and thus give a better representation of the data points. Here, K is the kernel chosen to suit the data. In this analysis the gaussian kernel has been chosen:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}. \tag{3.5}$$

The remaining items in the feature function are N_{tr} the number of training events, and $h_j = h^0 \times \sigma_j$, where σ_j is the standard deviation of variable j (in the space in which the first order correlations vanish) and h^0 is a tunable parameter which must be optimized for the data set. h^0 is chosen such that the functions formed by the sum of the product kernels are smooth representations without losing information about the data.

Using a linearly independent set of testing data, and the feature functions (f_s, f_b), a discriminant function value ($D(\mathbf{x})$) can be found for each event in the testing sample, representing the likelihood of the event being signal or background. This discriminant function is simply;

$$D(\mathbf{x}) = \frac{f_s}{f_s + f_b}, \tag{3.6}$$

where f_s is the feature function for the signal and f_b is the signal function for the background. Then a single cut on the value of $D(\mathbf{x})$ can be made, selecting the signal from the background by the value calculated for each event.

The PDE adaptive kernel builds on this method with one modification. An additional parameter α is used to further fit the gaussian kernels to the data set. A pilot f(x) is found using h_j and then the analysis is performed using;

$$h_j = \frac{h^0 \times \sigma_j \times f^{\alpha}_{pilot}(\overline{x})}{f^{\alpha}_{pilot}(x)}, \tag{3.7}$$

where α is another free parameter which must be optimized along with h^0 for the data set, f_{pilot} is the feature function formed using $h_j = h^0 \times \sigma_j$, and $\overline{x}$ is a vector of the mean values of each of the variables chosen for the analysis. This new choice of the width of the product kernels ties the functional form closer to the actual distribution of the data. For sets of data which may have a few outlying events, this now selects a wider gaussian, so that the feature function is smoother for these areas.

3.2.2 Implementation

The PDE method, as used in this analysis is exactly as described above in implementation. The only approximation made in the use of the technique is in the method used to diagonalize the covariance matrix of signal and background. For this procedure, a set of Jacobian rotations of the covariance matrices are made to find the eigenvalues and eigenvectors. These rotations are made until the sum of the off diagonal elements of the matrix are sufficiently small. The floating point precision of the machine determines how small the sum of the off diagonal elements must become. For a detailed description of these rotations and the algorithms the reader is referred to [8]. Once the eigenvalues are found, then one can form a transformation matrix which can be used to rotate the covariance matrices such that the first order correlations in the signal and background covariance matrices vanish. It can be shown that the matrix that achieves this transformation is;

$$A = v^T \times M, \tag{3.8}$$

$$M = (U\Lambda^{-\frac{1}{2}}U^T), \tag{3.9}$$

where U is the matrix of the eigenvectors of σ_b (the background covariance matrix), $\Lambda^{-\frac{1}{2}}$ is a diagonal matrix with one over the square root of the eigenvalues of σ_b, and v is the matrix of the eigenvectors of $M \times \sigma_b \times M$. As a consequence of this rotation, the diagonal elements of the background covariance matrix become ones. A proof of how this rotation may be performed can be found elsewhere [7]. This rotation may then be performed on the inputs, which results in the PDE analysis being performed in a space that is a linear combination of the input variables. The PDE algorithms have been implemented in C++ code, and prepared in a shared library form for use in ROOT (object oriented data analysis framework). In this analysis ROOT version 2.25/00 was used.

3.2.3 Simple examples of PDE

A few examples of PDE are presented here. These examples are meant to help the reader understand the action of the PDE method in a number of dimensions that can be visualized. The first is a one dimensional case in which there is separation between the sets of signal and background data. The data was created using a random number generator to supply a flat distribution of real numbers between [0,3] for signal and [4,7] for background. The histograms of these are shown in 3.2. Note that these two histograms have different scales and axis limits. The PDE analysis was performed

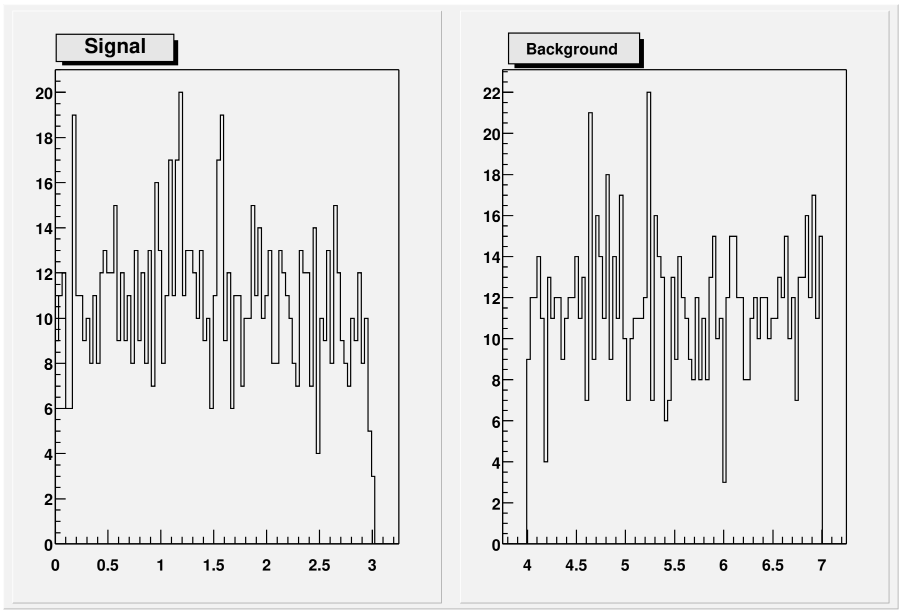

Figure 3.2 PDE 1-dimensional example data: Well separated data.

on this data in ROOT. The ROOT environment allows scripts to be written in C++ code. An example of the scripts used is shown in Appendix B (the macro that does the analysis for the tau lepton data set). The basic steps to using the PDE implementation in ROOT follow:

1. Load the shared library (pde.so) containing the PDE object and code into ROOT.

2. Create a pde object.

3. Create a list of TStrings that are the names of the variables that the analysis is to be performed upon.

4. Associate the TTrees that contain the signal and background data with the pde object (pde::SetTrees(TTree *signal, TTree *background)).

5. Set the PDE parameters, α and h^0 (pde::SetGlobals(float,float)).

6. Associate the list with the pde object (pde::BuildArrays(TList)).

7. Form the signal and background functions using either the adaptive kernel method or the fixed kernel (for fixed pde::Do(), for adaptive pde::DoAdaptive())

The analysis is essentially complete at this point. The signal function has been formed by a sum of gaussians about each of the points in the signal data. The background function has been created in a similar way, and as previously detailed the discriminant function D is formed. Presented in Figure 3.3 is a scatterplot of the discriminant function D versus the value of the variable x. PDE correctly selects a cut at x=3 for this data set, selecting only signal events.

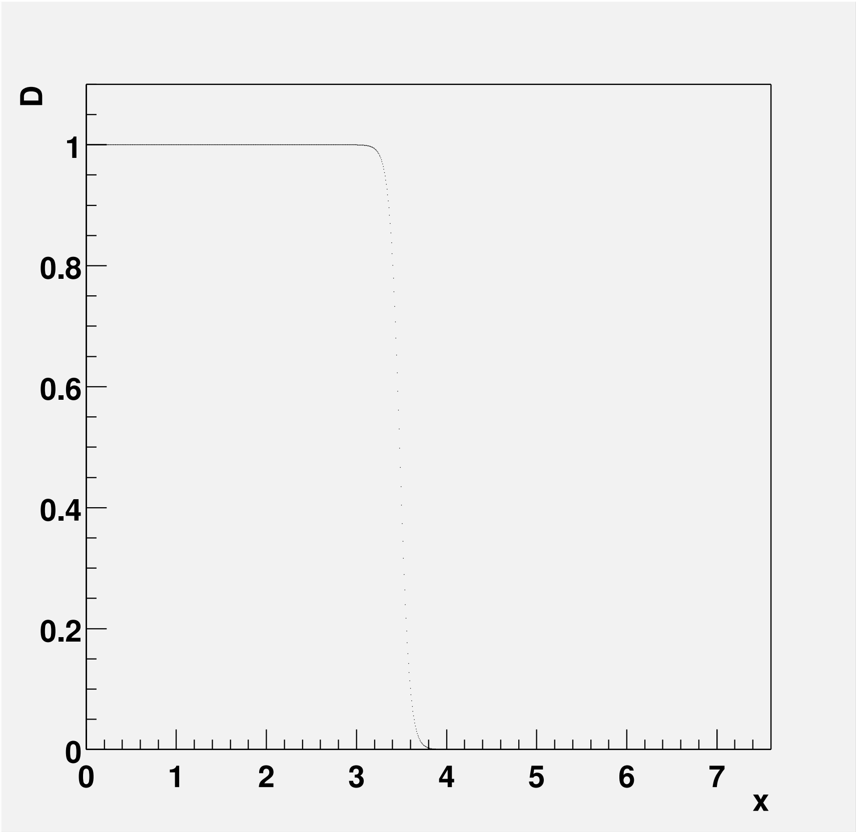

Figure 3.3 PDE 1-dimensional example result: Well separated data

The second example is similar to the first, except that there is a region of overlap between the one dimensional distributions. The signal data still runs from [0,3], but now the limits of the background are [2,5]. These distributions are shown in Figure 3.4. As before, the discriminant function is plotted as a function of the variable x so that one can see where the signal is the most likely. (in Figure 3.5). The analysis now shows

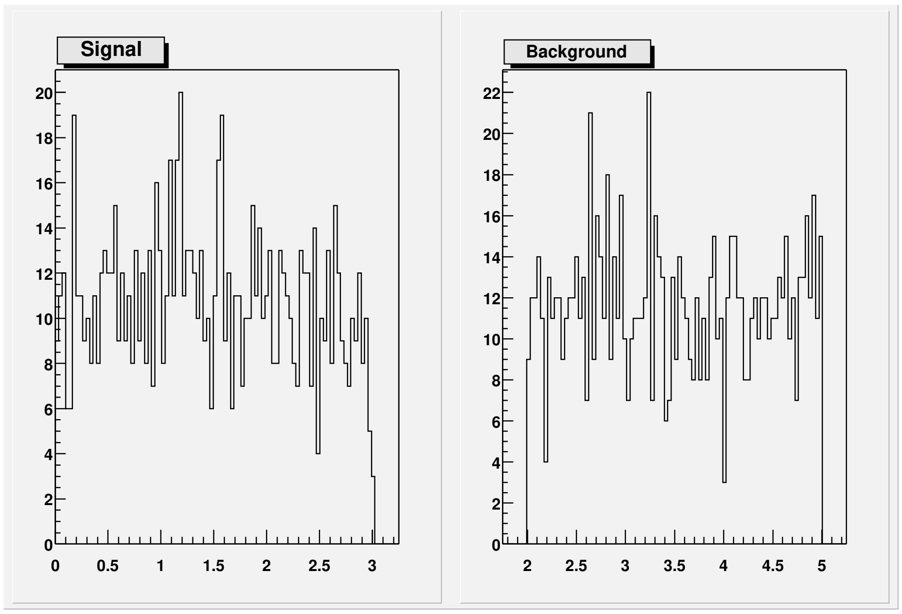

Figure 3.4 PDE 1-dimensional example data: Data with overlap

that PDE is certain that all of the data for x<2 is signal, and assigns those values a discriminant of 1. The values that show an overlap are assigned an intermediate value, due to the fact that the probability of the value being signal or background is the same in this area. At x=3, the discriminant trails off and falls to zero since there is no signal data beyond this point.

As a more complicated example, a two dimensional case is considered. The signal is a decaying exponential in the x-direction, and a gaussian of width 0.15 in the y-direction. The background is a two dimensional gaussian centered about (0.6, .25) with widths of (0.35,0.55) respectively. Lego plots of these distributions are shown in Figure 3.6. Performing the PDE analysis on this data set, the feature functions are

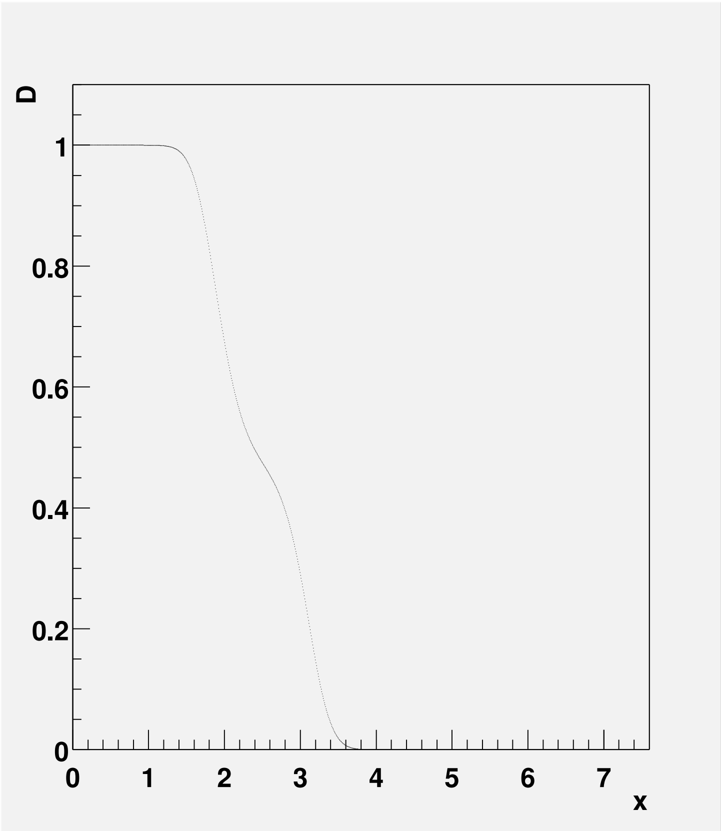

Figure 3.5 PDE 1-dimensional example result: Data with overlap

formed by a product of gaussians (one for each dimension) and then summing these two-dimensional gaussians to create a function of the two input variables instead of the one variable case previously discussed. As before, the discriminant function is plotted as a function of the two input variables (Figure 3.2.3).

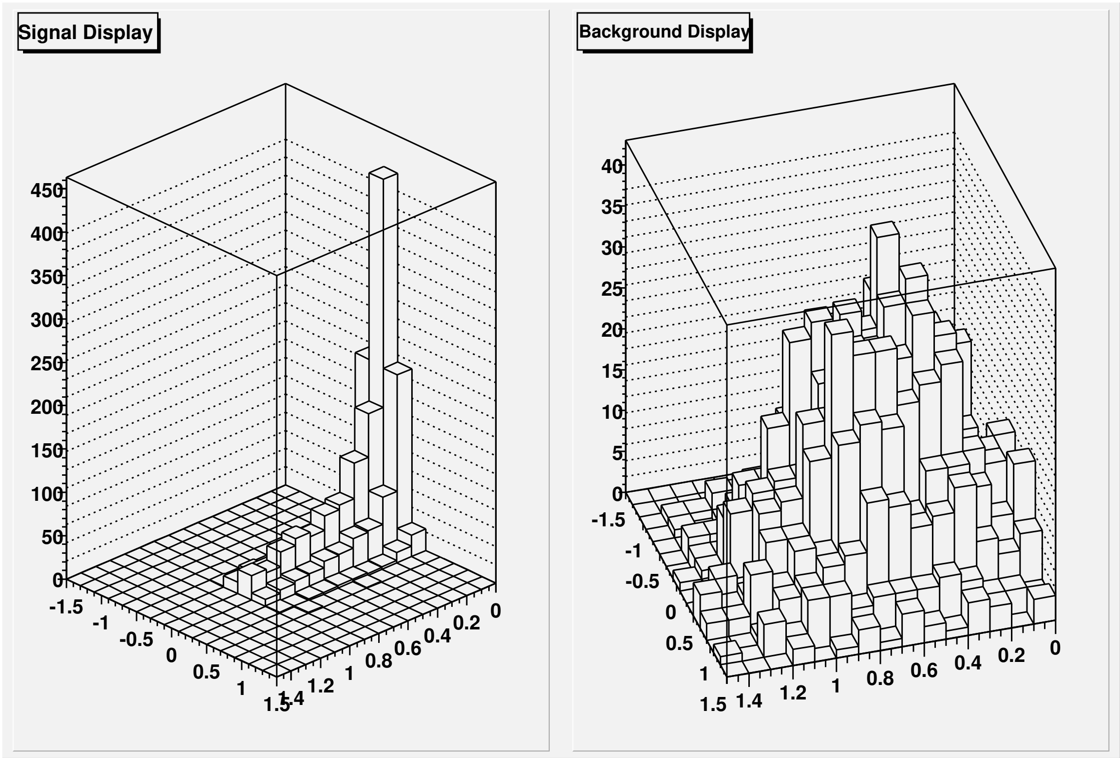

Figure 3.6 PDE 2-dimensional example data

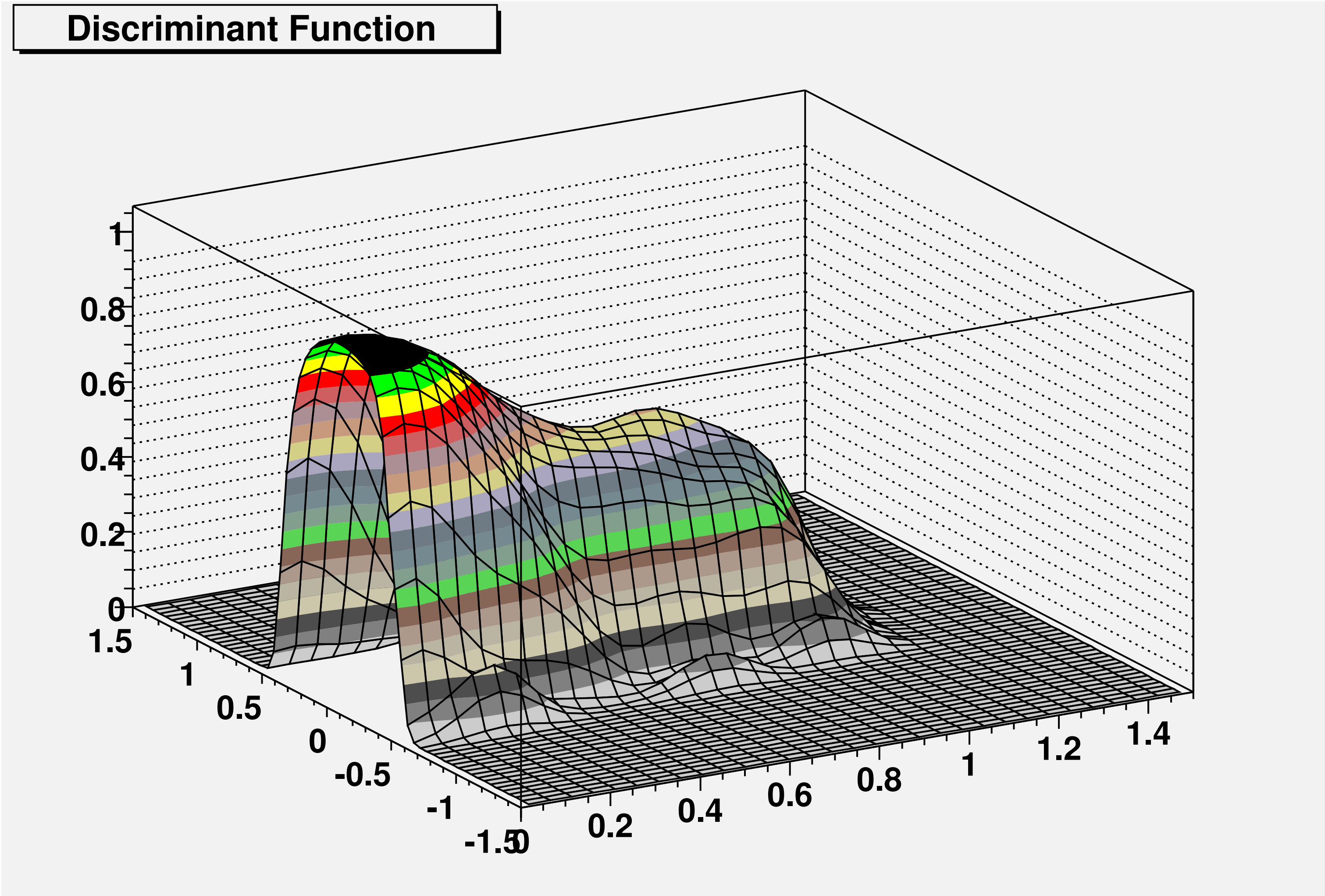

Figure 3.7 PDE 2-dimensional example result

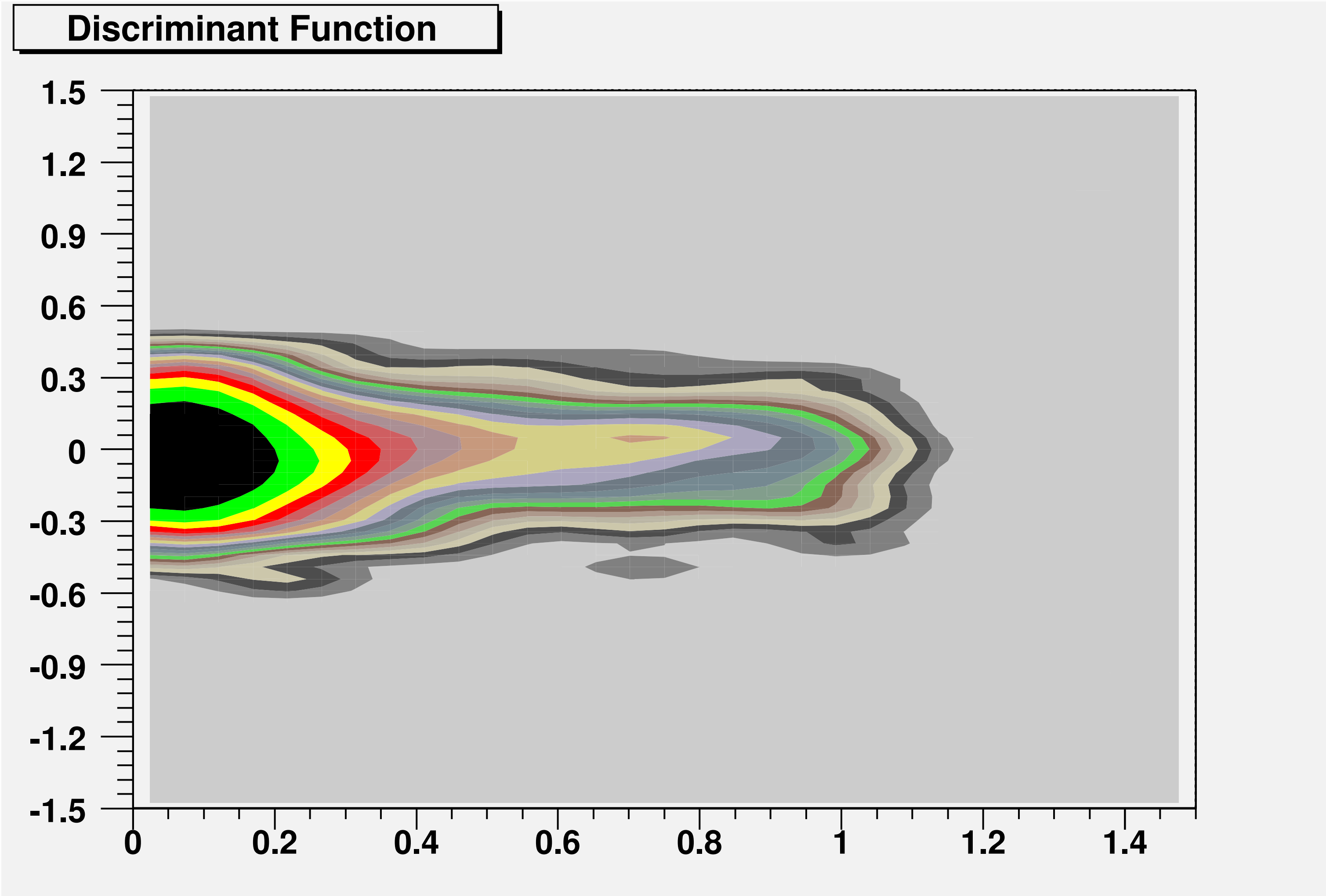

Figure 3.8 PDE 2-dimensional example result, Contour plot.

Chapter 4
Analysis

The data samples selected for analysis are Monte Carlo simulations of tau lepton and electron events in the previously described DØ detector. These events were generated using the PYTHIA Monte Carlo generator. The generator provides the kinematic properties of the particles from the process being examined. The information from the generator was then input to DØGSTAR. DØGSTAR is another Monte Carlo program, the DØ GEANT Simulation of Total Apparatus Response (GEANT being the standard in detailed simulation software for high energy physics). The outputs from DØGSTAR are in the form of raw quantities that would be read out directly from the detector instrumentation. This data is then input to the DØ reconstruction program which then constructs detector objects such as calorimeter clusters and tracks. The detector coordinates of the centers of calorimeter clusters are calculated here. The efficiencies quoted in the analysis do not take into account trigger efficiencies or online selection criteria that have occurred in this reconstruction.

Several conventions are used that require definition. The calorimeter is divided into towers as described in the detector section. Each tower is composed of cells, which are the output from devices that measure the signal from a particle shower (smallest division in the calorimeter). The energy deposited in each cell is calculated from this signal. From a cell's coordinates within the detector and the interaction vertex, a vector is formed. Using this vector, the transverse energy is calculated;

$$E_T = E sin\theta, \tag{4.1}$$

where E is the energy deposited in the cell and θ is the angle that the vector from the interaction vertex to the cell makes with the beam line. The transverse energy of a given tower then is the sum of the transverse energies of its component cells.

A momentum is also assigned to calorimeter cells, assuming the high energy limit (E $\gg$ m). The transverse momentum is defined as the transverse energy above, and the vector sum of the momentum of the cells in a tower is the tower's transverse momentum.

For the identification of single electrons, two cases of backgrounds are considered. The first was QCD jets with a significant amount of their energy deposited in the electromagnetic layers of the calorimeter. The second was a charged pion, to provide the necessary track, and an uncharged pion, to provide the electromagnetic shower via its decay to two photons, traveling at close angle to each other.

For tau events, only the hadronic decays are examined. The leptonic decays can be separately identified by the identification of the accompanying electron or muon, with the requirement of sizeable missing transverse energy to account for the two neutrinos. The background for the tau events are QCD jets.

Each data set for the analysis was in HBOOK (PAW) format for the MLPfit neural network analysis. Then the files were converted into ROOT format, and the PDE analysis was performed. The same variables and same events in each analysis were used in each technique. For the direct comparison of the two techniques, the weights from the neural networks trained with MLPfit were copied into a C++ macro so that the neural network results could be plotted together with the PDE results in ROOT.

4.1 Single Electron Analysis

4.1.1 Single Electron Identification Data Sets

All of the single electron analysis data sets were subjected to the same selection cuts. The following criteria were required:

1. Events were required to have $\mathrm{EM}_{fraction}$ >0.65, at least 65% of the total energy of the particle or jet deposited within the electromagnetic layers of the calorimeter.

2. Acceptable events were required to be in the energy range of 10< E_{tot} <100 GeV.

3. Events were required to be in the central calorimeter or $|\eta| < 1.1$.

4. The calorimeter cluster was required to have σ_{ϕ} <0.25, where σ_{ϕ} is defined below in the description of the variables used in the analysis.

As previously noted, two backgrounds were considered. In the first case, the background used was QCD jets that met the above characteristics. The second case of two pions was studied by choosing the decay of $\tau \to \nu\pi\pi^0$ with subsequent $\pi^0 \to \gamma\gamma$ as the background. Table 4.1 summarizes the total events and the corresponding number of events surviving the cuts.

	Total Number of events:	Events after cut:
Single electron vs. $\tau \to \pi^0 \to \gamma\gamma$		
Signal Training:	1400	1256
Signal Testing:	1498	1344
Background Training:	2200	1640
Background Testing:	2257	1693
Single electron vs. QCD jets		
Signal Training:	4844	2593
Signal Testing:	4746	2458
Background Training:	4276	1581
Background Testing:	4756	1808

Table 4.1 Events Selected for Single Electron Analysis

4.1.2 Single Electron Versus QCD jets Variables

For the discrimination between electrons and QCD jets, seven variables were selected. These were the current set of variables being studied by neural network analysis when this comparison was performed. The electromagnetic calorimeter is emphasized, since all of the events were required to have 65% of their energy deposited there. The following variables were selected to represent the features of the signal (histograms of the training sets, normalized by the number of events follow the variable definitions):

1. E_{em1}/E_{tot}: The sum of the energy deposited in the first layer of the electromagnetic calorimeter divided by the total energy of the calorimeter cluster.

2. $\log_{10} E_{tot} - 1$: Logarithm (base 10) of the total energy of the calorimeter cluster minus one.

3. E_{em3}/E_{tot}: Energy deposited in the third layer of the electromagnetic calorimeter divided by the total energy of the calorimeter cluster.

4. $(E_{em1} + E_{em2} + E_{em3} + E_{em4} + E_{fh1})/E_{tot}$: The sum of the energy deposited in the four layers of the electromagnetic calorimeter and the first layer of the hadronic calorimeter divided by the total cluster energy.

5. E_{fh1}/E_{tot}: Energy deposited in the first finely segmented layer of the hadronic

calorimeter divided by the total energy of the calorimeter cluster.

6. Z_{vert}: The position of the vertex in the z-direction. In this case there is no normalization attached to this quantity.

7. Cluster width: Square root of the sum of the squares of the eta and phi widths of the calorimeter cluster.

$$\sqrt{(\sigma_\phi)^2 + (\sigma_\eta)^2} \tag{4.2}$$

$$\sigma_\phi^2 = \frac{\sum_i (\phi_i - \phi_C)^2 E_{T_i}^2}{\sum_i E_{T_i}^2}, \sigma_\eta^2 = \frac{\sum_i (\eta_i - \eta_C)^2 E_{T_i}^2}{\sum_i E_{T_i}^2}, \tag{4.3}$$

where (η_i, ϕ_i) are the coordinates of the ith calorimeter tower, (η_C, ϕ_C) are the coordinates of the calorimeter cluster, and E_{T_i} is the transverse energy of the tower.

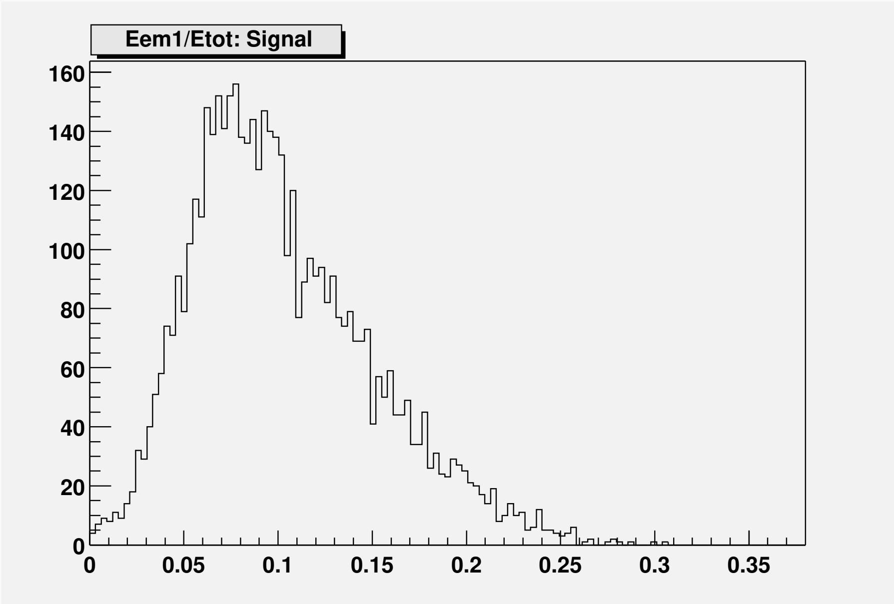

Figure 4.1 E_{em1}/E_{tot} for signal

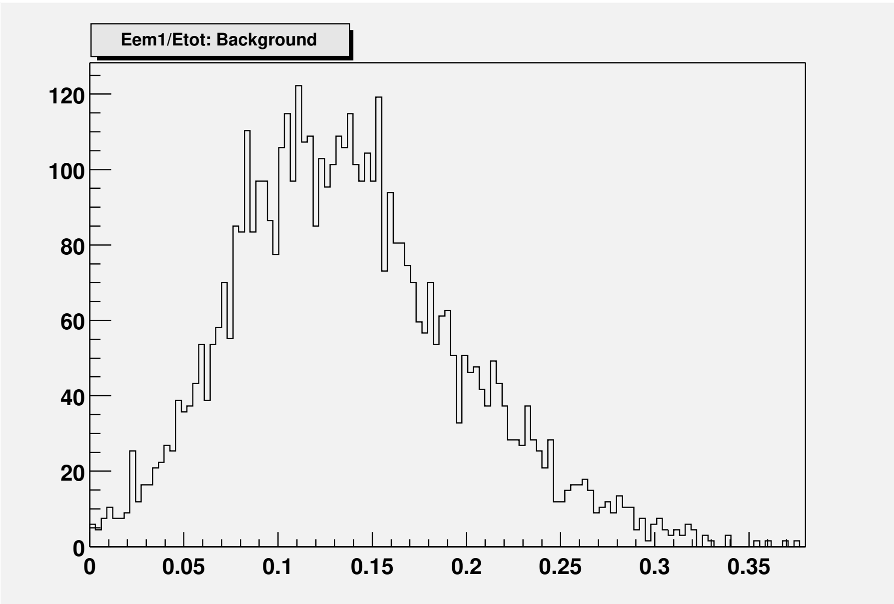

Figure 4.2 E_{em1}/E_{tot} for background

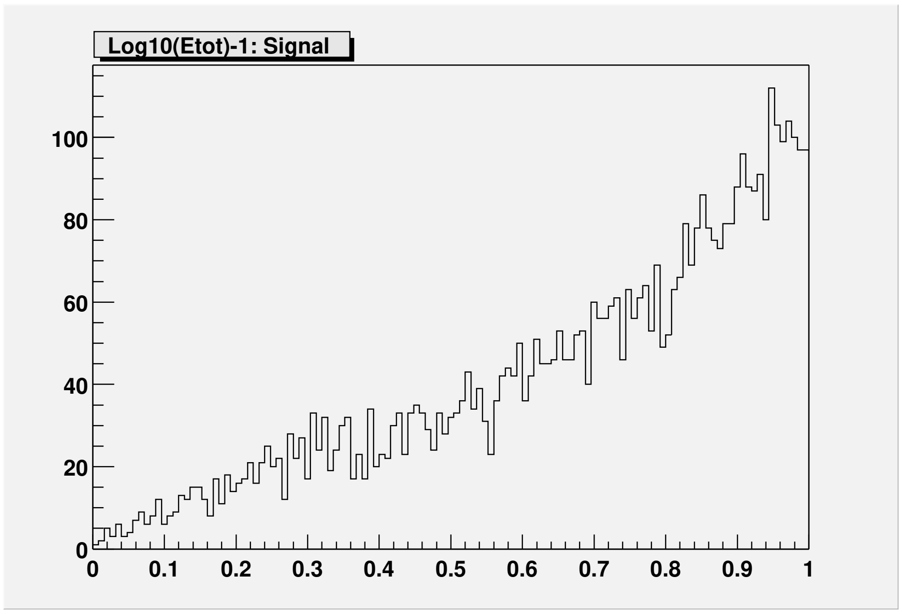

Figure 4.3 $\log_{10} E_{tot} - 1$ for signal

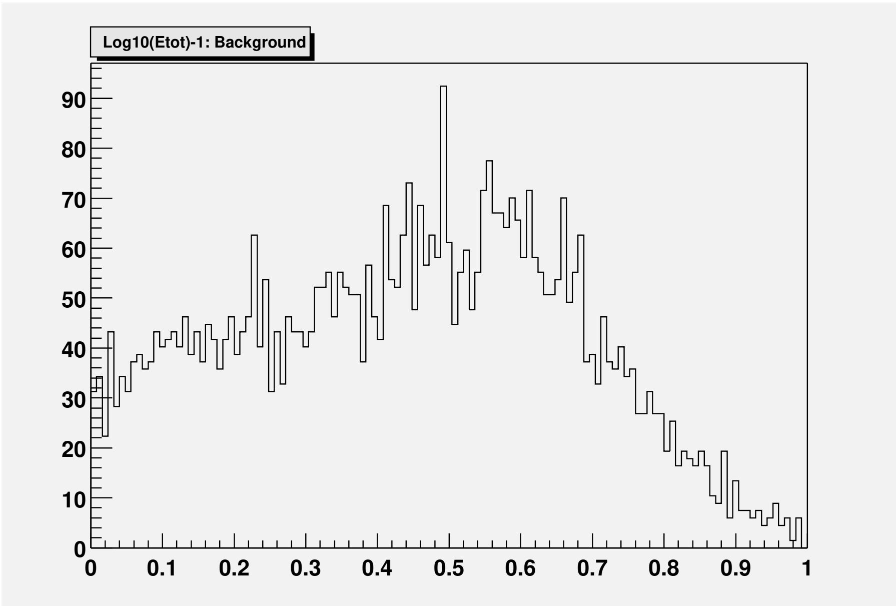

Figure 4.4 $\log_{10} E_{tot} - 1$ for background

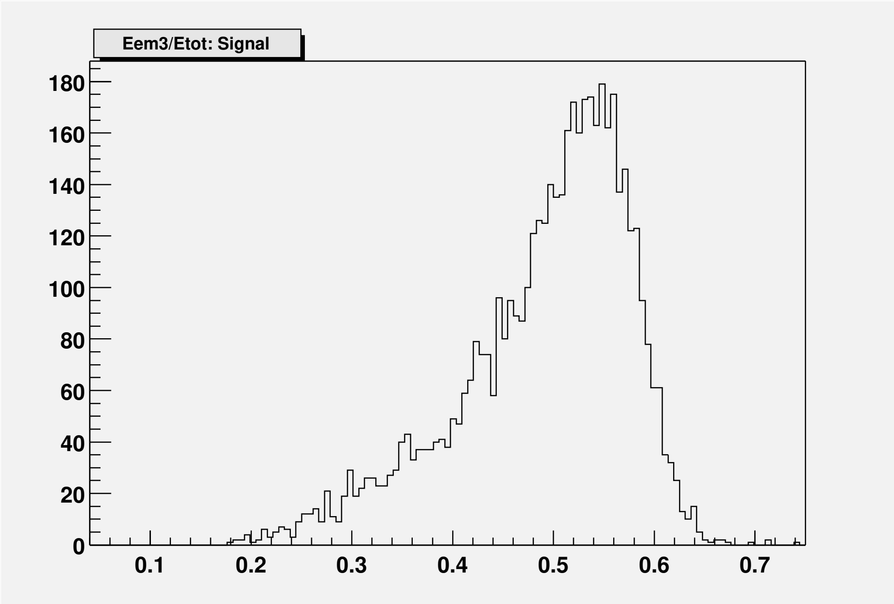

Figure 4.5 E_{em3}/E_{tot} for signal

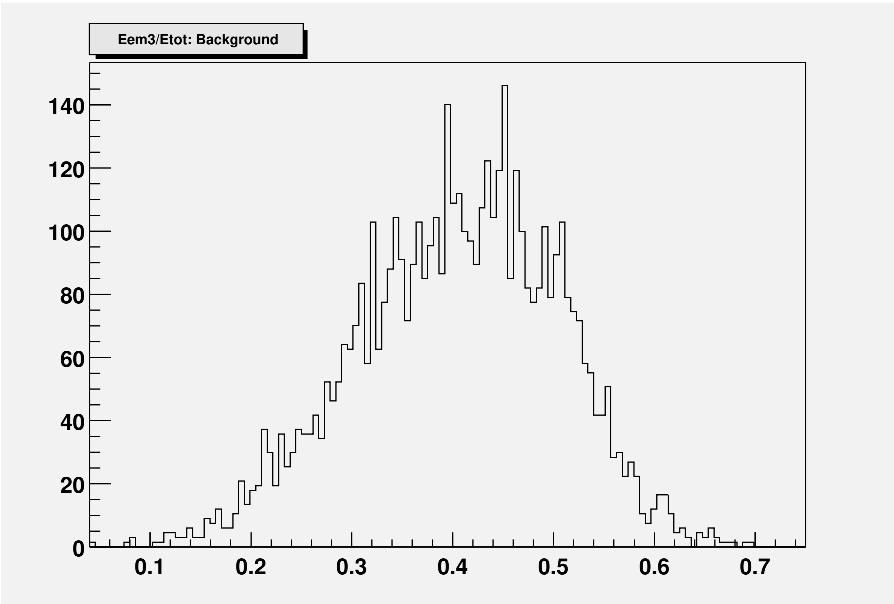

Figure 4.6 E_{em3}/E_{tot} for background

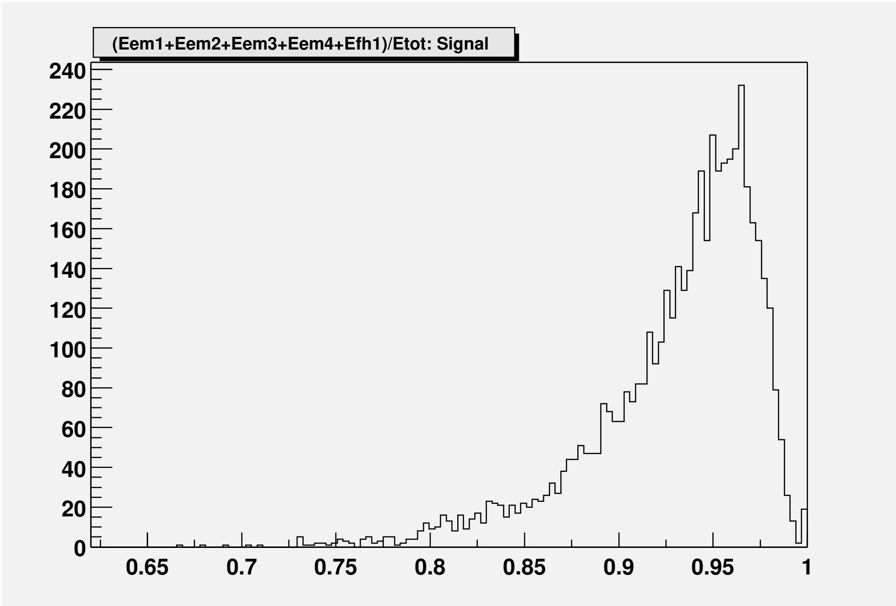

Figure 4.7 $\frac{(E_{em1}+E_{em2}+E_{em3}+E_{em4}+E_{fh1})}{E_{tot}}$ for signal

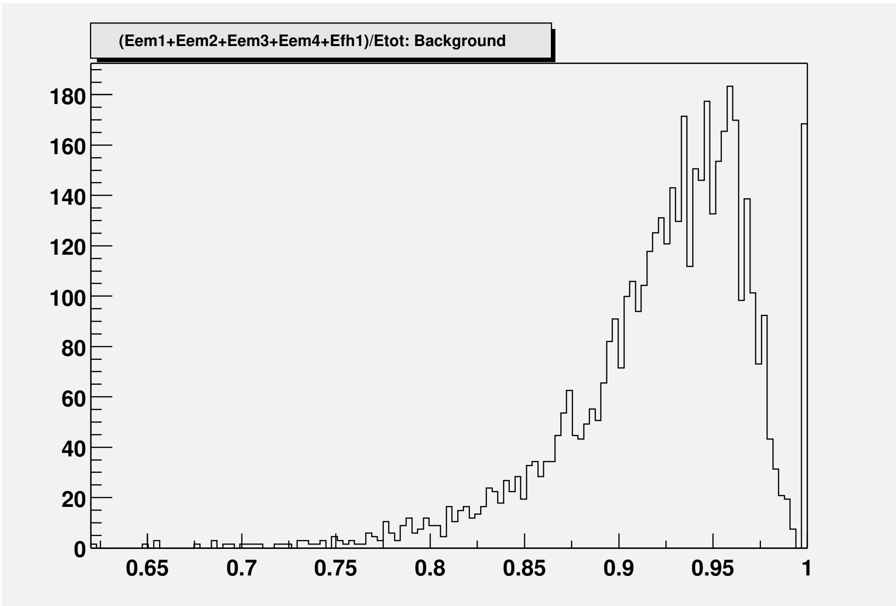

Figure 4.8 $\frac{(E_{em1}+E_{em2}+E_{em3}+E_{em4}+E_{fh1})}{E_{tot}}$ for background

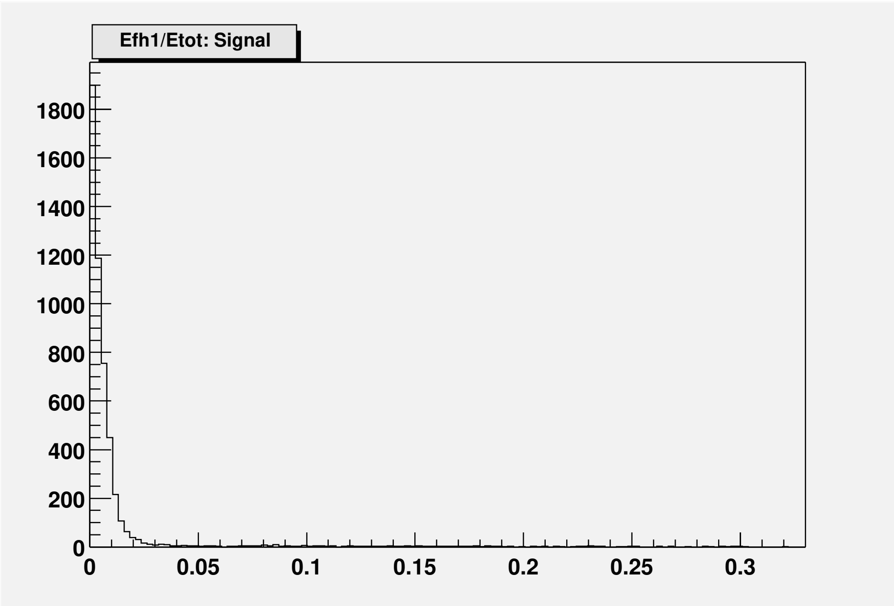

Figure 4.9 E_{fh1}/E_{tot} for signal

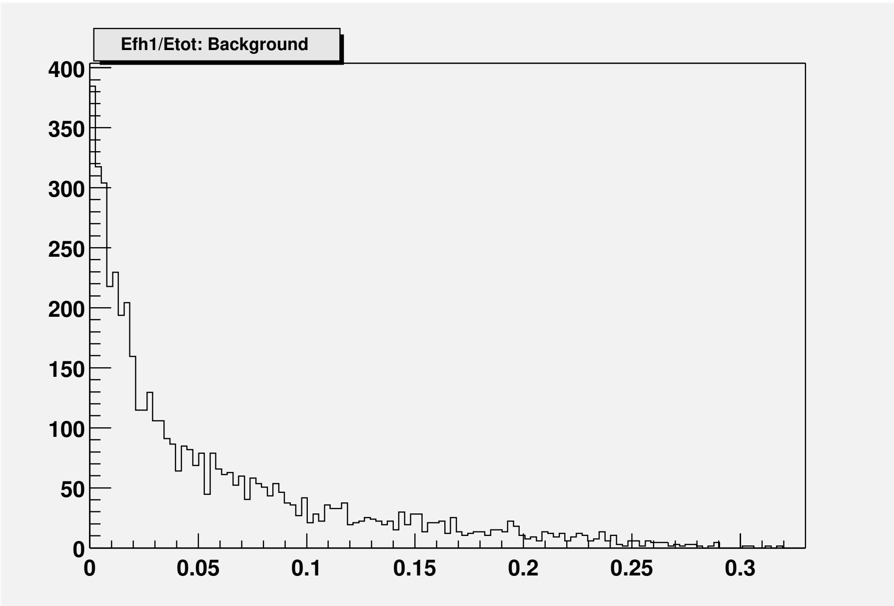

Figure 4.10 E_{fh1}/E_{tot} for background

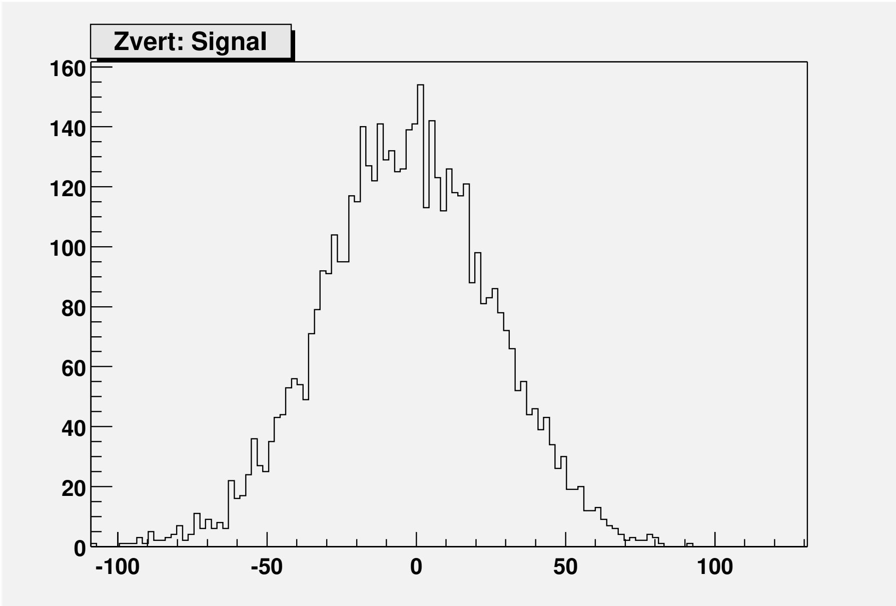

Figure 4.11 Z_{vert} for signal

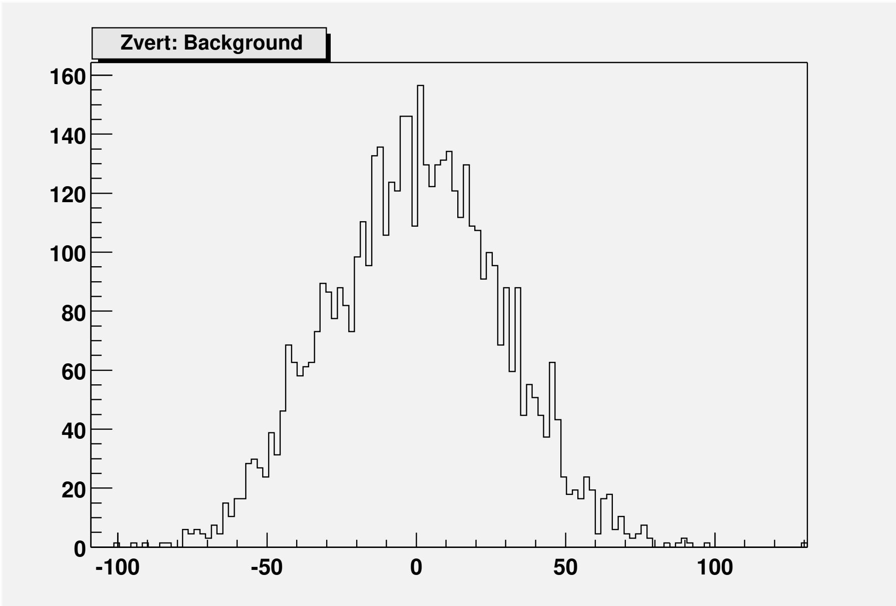

Figure 4.12 Z_{vert} for background

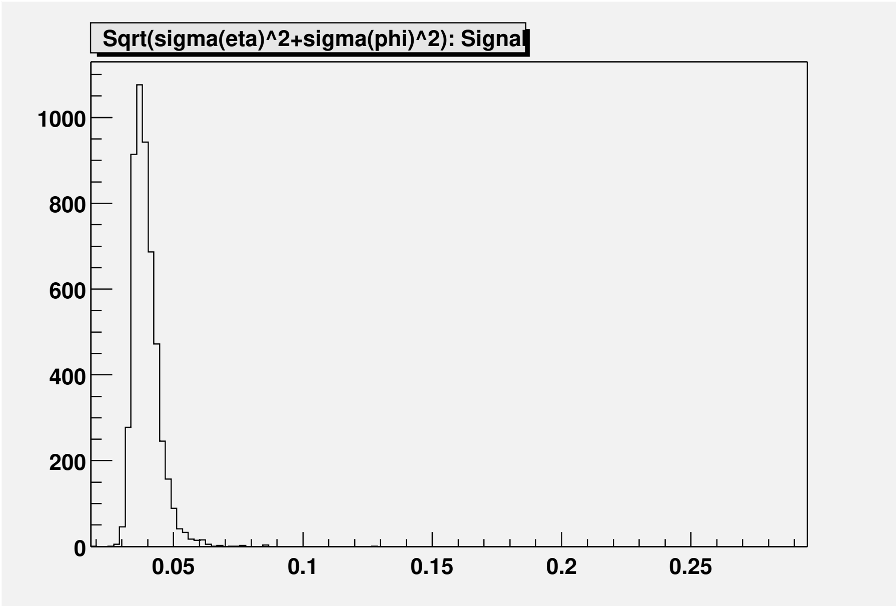

Figure 4.13 Cluster width for signal

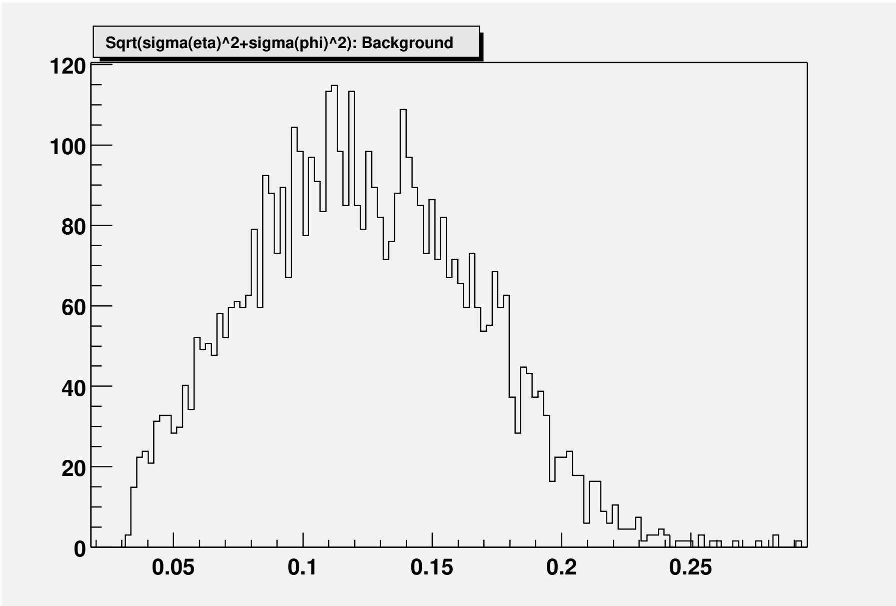

Figure 4.14 Cluster width for background

4.1.3 Single Electron Versus $\tau \rightarrow \pi^0 \rightarrow \gamma\gamma$ Variables

The variables chosen for the analysis to discriminate between electrons and $\tau \rightarrow \pi^0 \rightarrow \gamma\gamma$ are nearly the same as those used in the previous section, though some normalizations differ. As before, they were the variables currently being used in both Hmatrix and neural network studies when this comparison was performed. Histograms of the training data sets, normalized by the number of variables follow the definitions. These seven variables were:

1. $(E_{em1} + E_{em2} + E_{em3} + E_{em4} + E_{fh1})/E_{tot}$: The sum of the energy deposited in the four layers of the electromagnetic calorimeter and the first layer of the hadronic calorimeter divided by the total cluster energy.

2. E_{fh1}/E_{tot}: Energy deposited in the first finely segmented layer of the hadronic calorimeter divided by the total energy of the calorimeter cluster.

3. $(E_{em1}+E_{em2})/E_{tot}$: The sum of the energy deposited in the first two layers of the electromagnetic calorimeter divided by the total energy of the calorimeter cluster.

4. E_{em3}/E_{tot}: Energy deposited in the third layer of the electromagnetic calorimeter divided by the total energy of the calorimeter cluster.

5. $\frac{Z_{vert}}{100}$: The position of the vertex in the z-direction divided by one hundred.

6. Cluster width, as defined in the previous section.

7. $\log_{10} E_{tot}$: Logarithm (base 10) of the total energy of the calorimeter cluster.

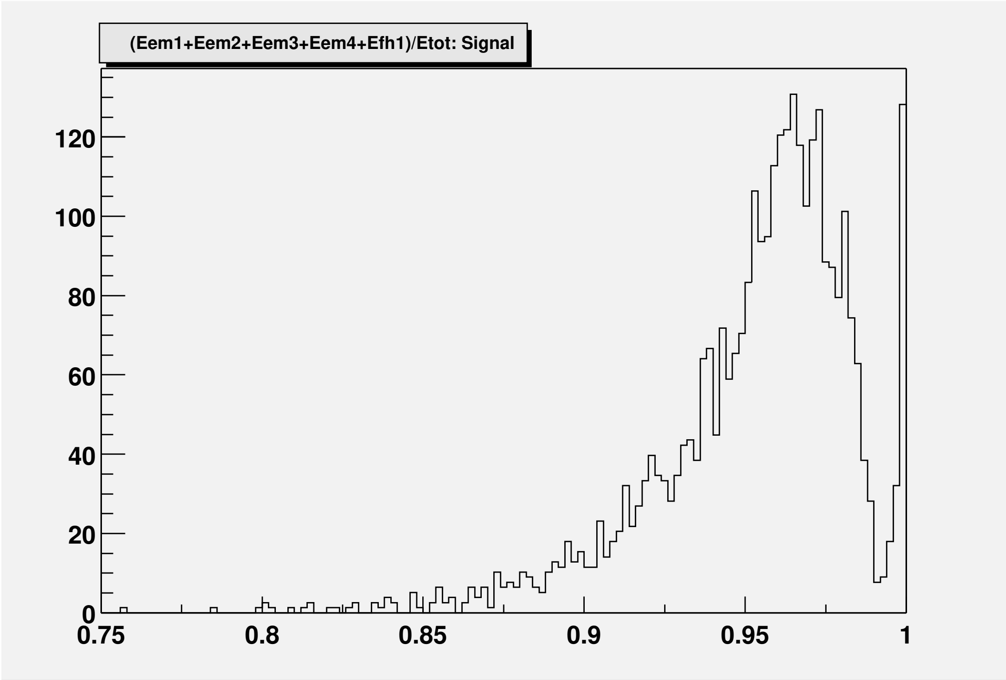

Figure 4.15 $\frac{(E_{em1}+E_{em2}+E_{em3}+E_{em4}+E_{fh1})}{E_{tot}}$ for signal

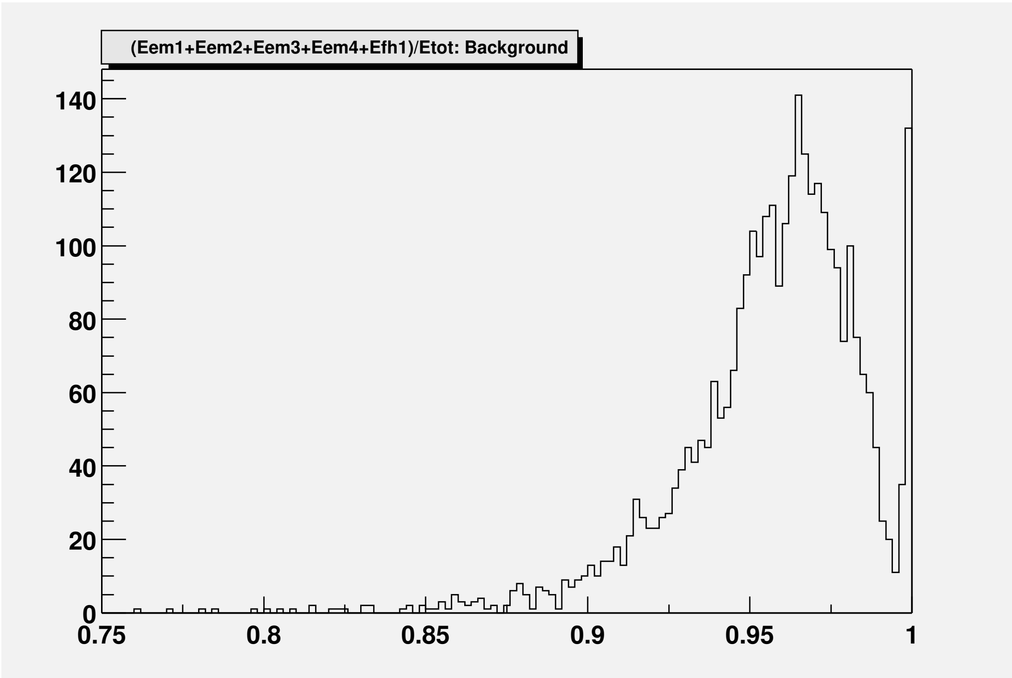

Figure 4.16 $\frac{(E_{em1}+E_{em2}+E_{em3}+E_{em4}+E_{fh1})}{E_{tot}}$ for background

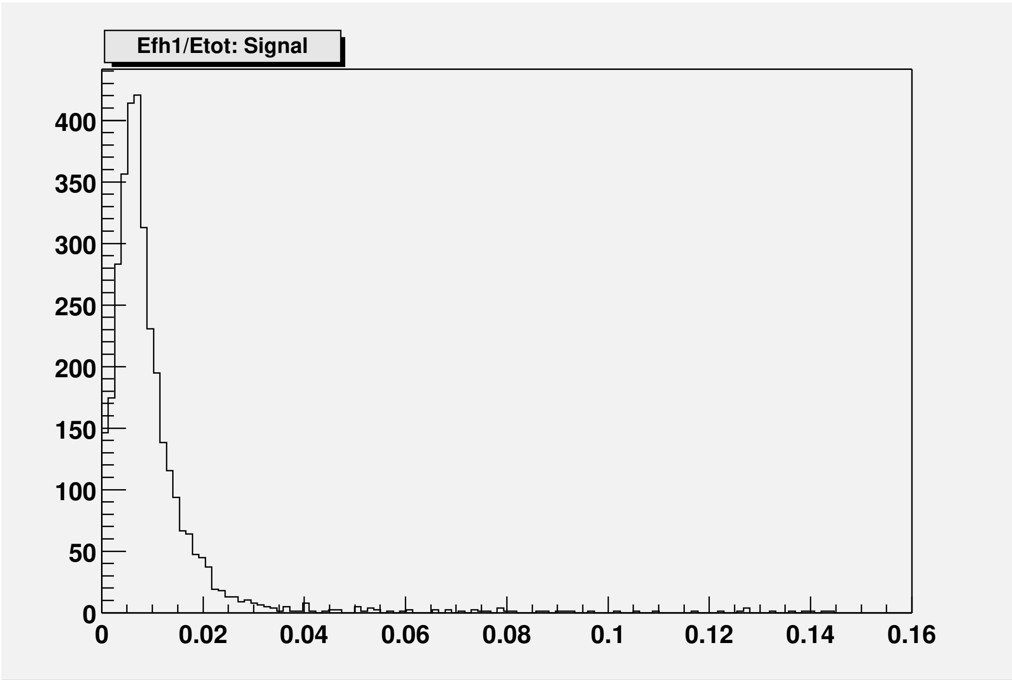

Figure 4.17 E_{fh1}/E_{tot} for signal

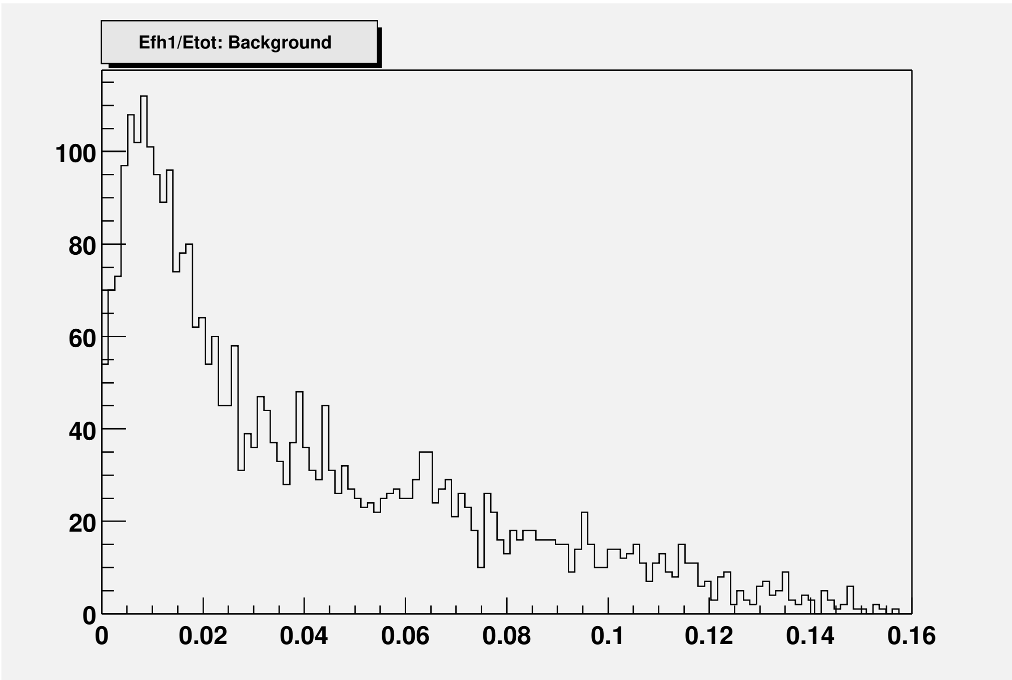

Figure 4.18 E_{fh1}/E_{tot} for background

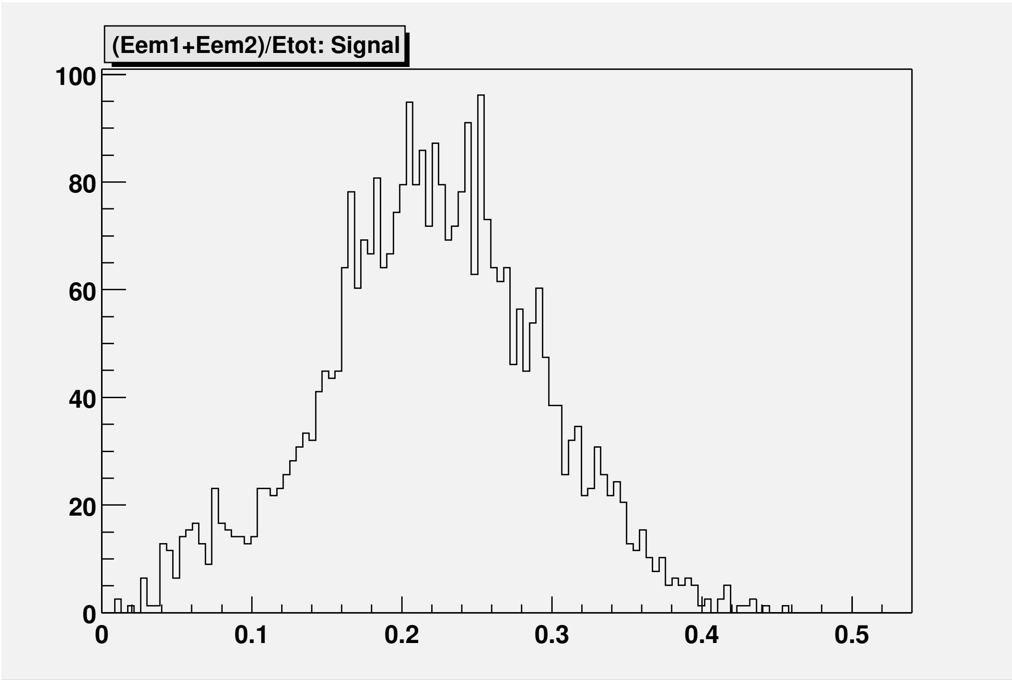

Figure 4.19 $(E_{em1} + E_{em2})/E_{tot}$ for signal

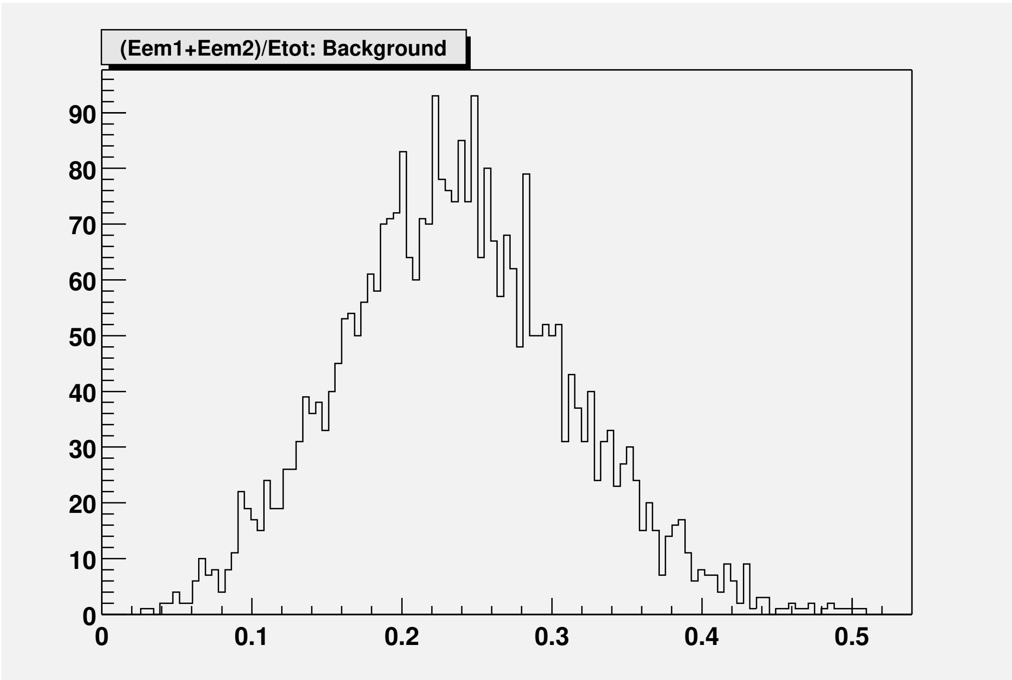

Figure 4.20 $(E_{em1} + E_{em2})/E_{tot}$ for background

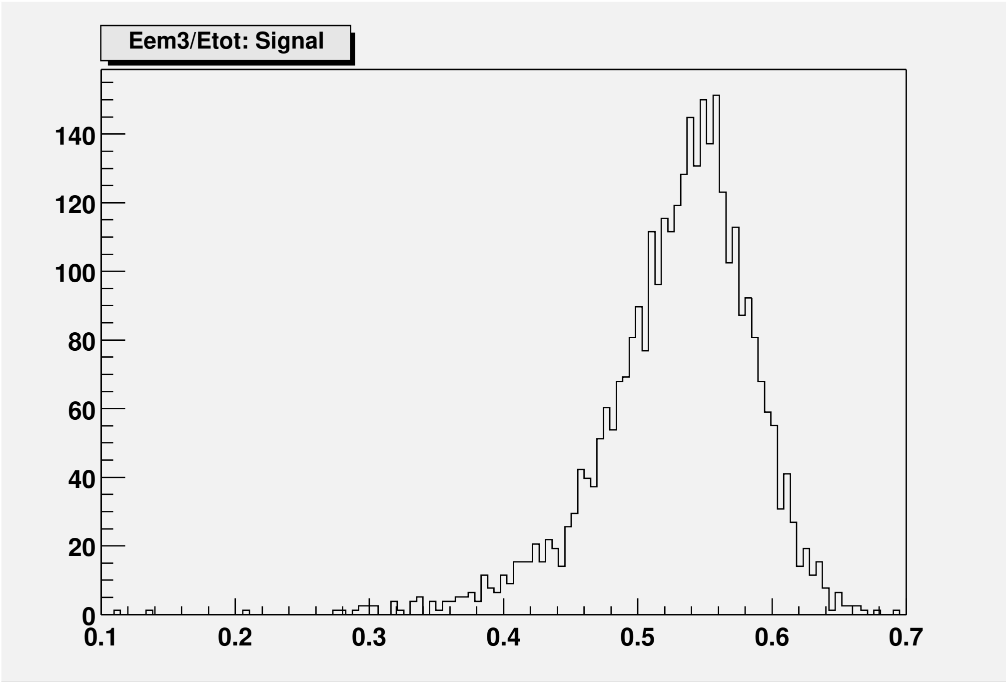

Figure 4.21 E_{em3}/E_{tot} for signal

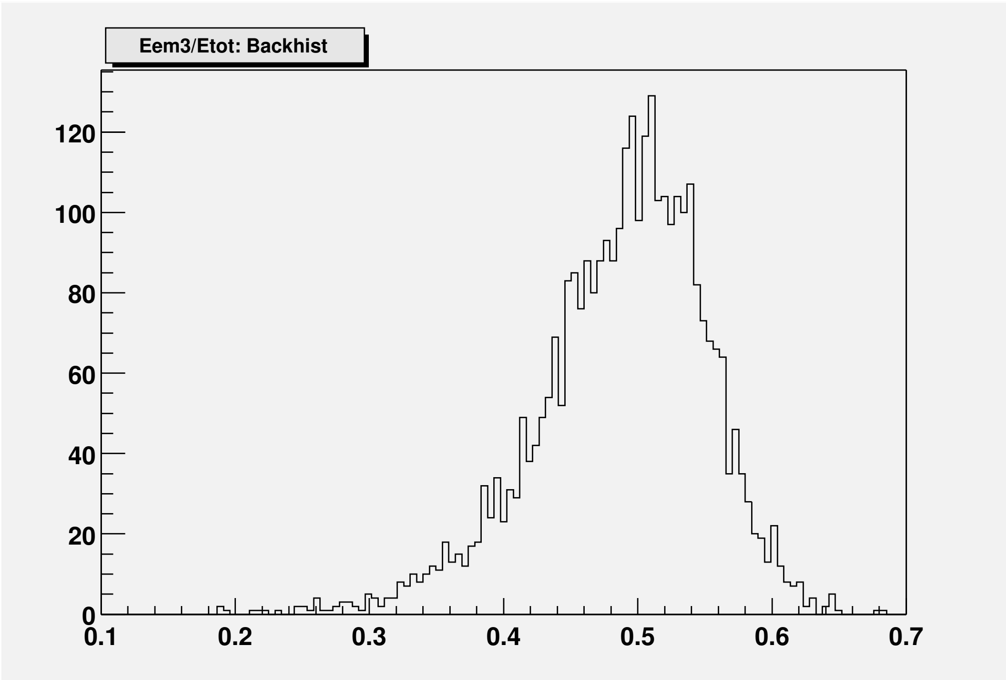

Figure 4.22 E_{em3}/E_{tot} for background

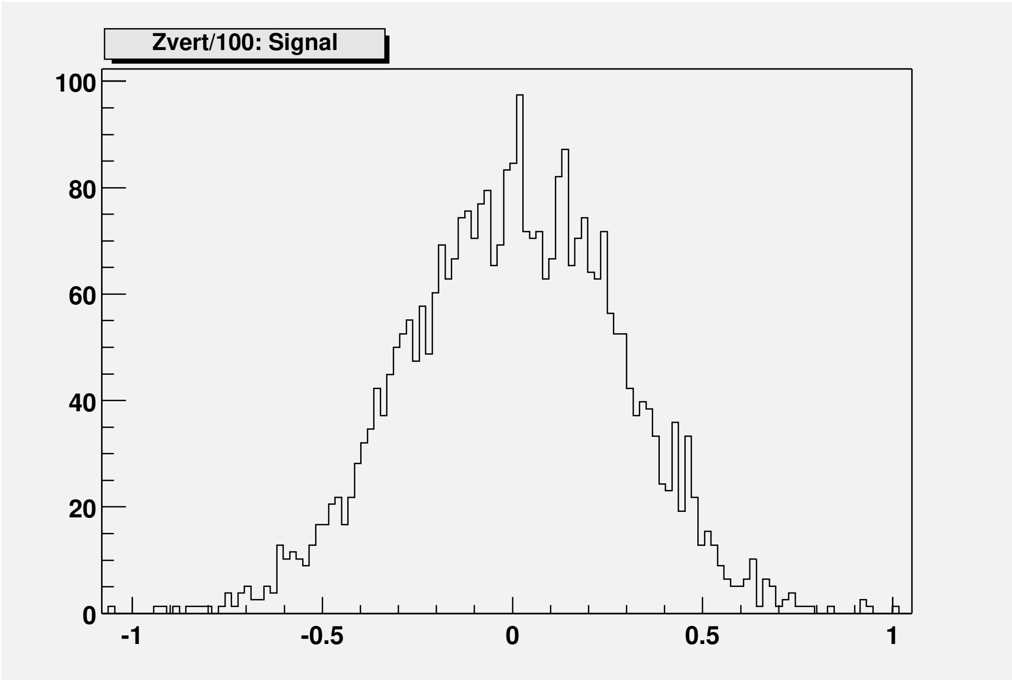

Figure 4.23 $\frac{Z_{vert}}{100}$ for signal

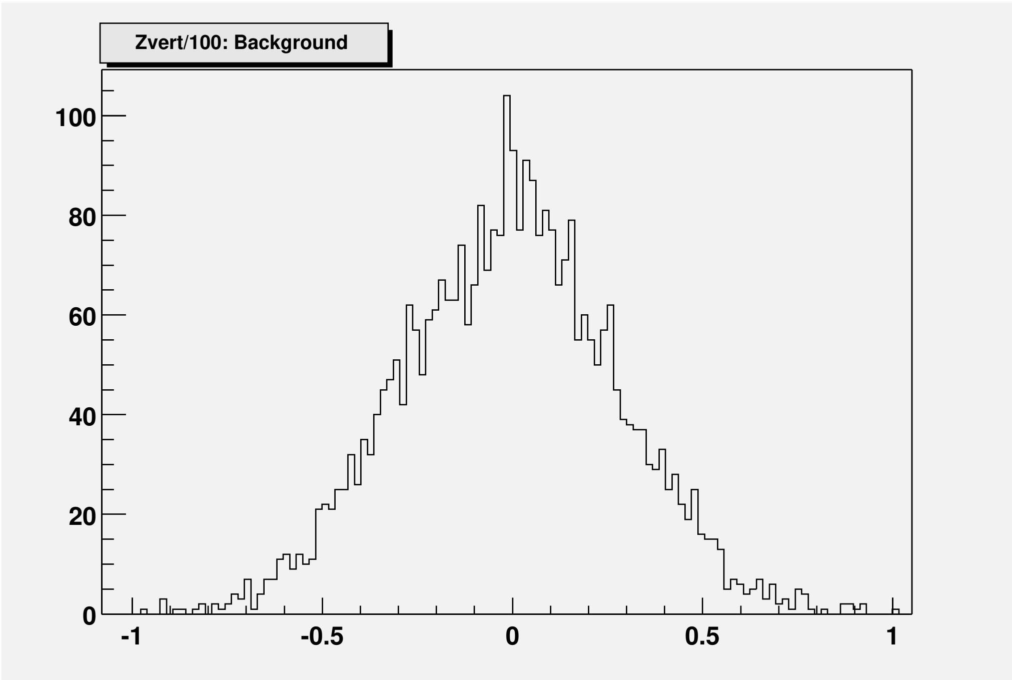

Figure 4.24 $\frac{Z_{vert}}{100}$ for background

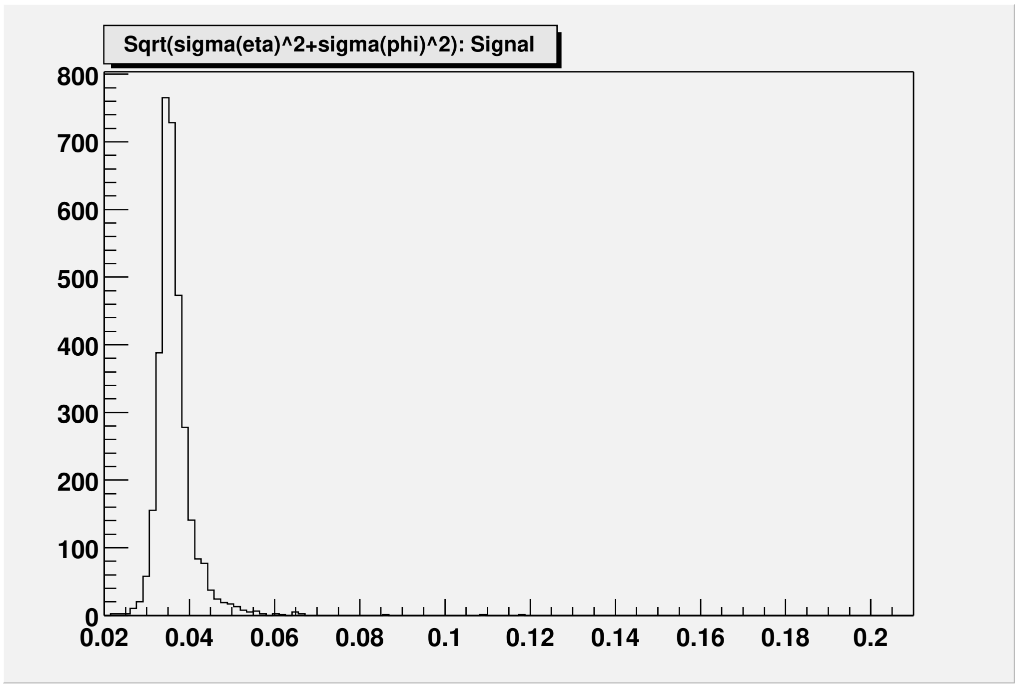

Figure 4.25 Cluster width for signal

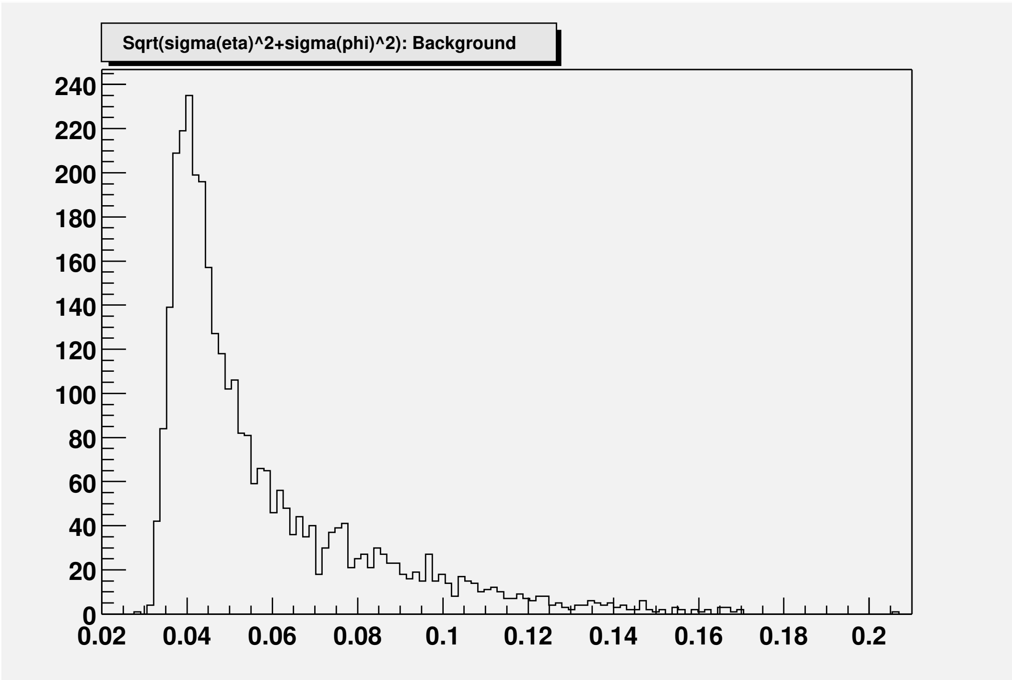

Figure 4.26 Cluster width for background

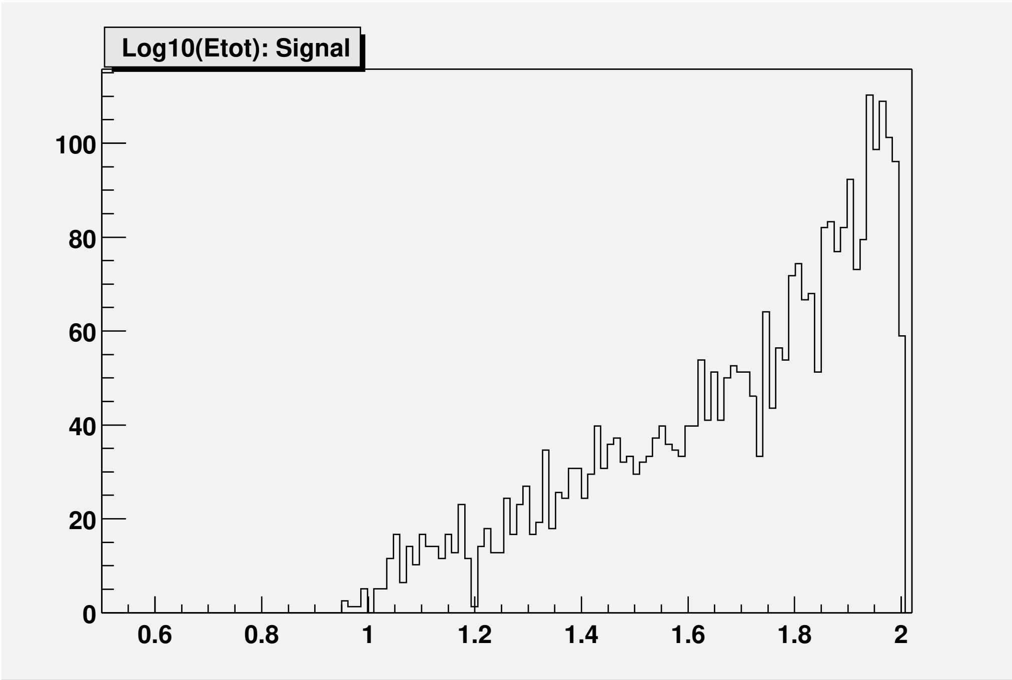

Figure 4.27 $\log_{10} E_{tot}$ for signal

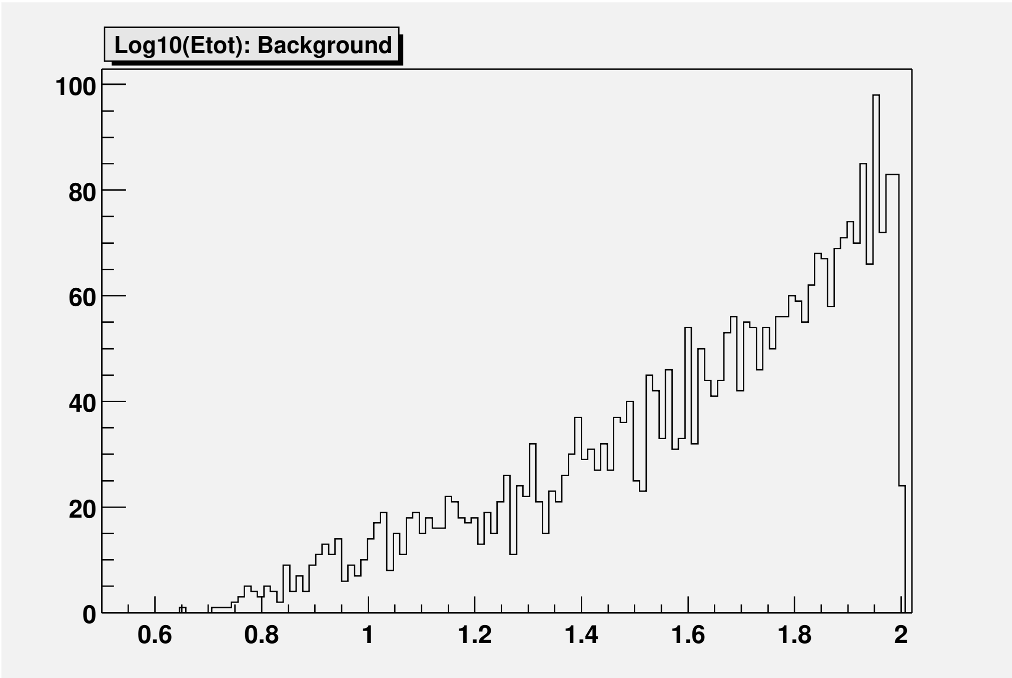

Figure 4.28 $\log_{10} E_{tot}$ for background

4.1.4 Optimization of Single Electron Neural Network Parameters

To ensure that the maximum efficiency in discriminating between signal and background is obtained, the parameters of the neural network were varied and the performance was studied. Before discussing the optimizations performed, the terms used need to be defined.

The signal efficiency is defined as the number of signal events in the testing sample that are included in a given cut on the neural network output divided by the total number of signal events.

$$\epsilon_s = \frac{N_{signal\ events\ after\ cut}}{N_{total\ signal\ events}} \tag{4.4}$$

The background efficiency is defined likewise. The output from neural networks are peaked about the specified values (1 and 0) for which the network was trained, however the distribution trails off to either side. When results of different networks were compared, the results were binned in histograms with the same number of bins, but axis limits that included all of the events. For the efficiency comparisons an efficiency was computed for each bin in the histogram and those efficiencies were compared.

The purity of the data is defined to be:

$$Purity = \frac{\frac{N_{signal\ events\ after\ cut}}{N_{total\ signal\ events}}}{\frac{N_{signal\ events\ after\ cut}}{N_{total\ signal\ events}} + \frac{N_{background\ events\ after\ cut}}{N_{total\ background\ events}}}. \tag{4.5}$$

The purity multiplied by the signal efficiency was the quantity studied to determine the performance of the network. This quantity is defined to be;

$$Purity \times \epsilon_s = \frac{\epsilon_s}{\epsilon_s + \epsilon_b} \times \epsilon_s, \tag{4.6}$$

where ϵ_s is the signal efficiency and ϵ_b is the background efficiency. This has a dependence on the background efficiency as well as a quadratic dependence on the signal efficiency indicating that there can be a maximum. The maximum purity times signal efficiency determines where the cut on the neural network should be to obtain the best performance. Note that on the graphs shown, Purity times signal efficiency has been abbreviated to Purity for brevity.

In each case, the default PAW parameter set was used to train the neural network (random set of starting weights, BFGS quasi-Newton learning method). The error of the neural network was displayed online in PAW as the training was conducted. Points along this error curve were chosen, and the performance at those points were compared. The performance was compared to ensure that enough training cycles were used to properly optimize the weights, and to ensure that the network was not overtrained. Neural networks that are overtrained are tuned so precisely to the sets of training data that independent samples of testing data are not properly characterized. The weights were saved online, so that only one neural network was trained, and the performance comparison reflects changes made by training. The optimum number of cycles was determined by finding the number of cycles for which the maximum purity times signal efficiency could be obtained, and past which subsequent training lowered this maximum. For each analysis, a constant number of nodes was used while the number of training cycles was changed. The graphs shown in Figures 4.29 and 4.30 are the purity multiplied by the signal efficiency versus a cut on the neural network output. The different lines represent the different numbers of cycles trained over. These graphs are shown to illustrate that selecting an arbitrary point to take the maximum purity times signal efficiency was not optimal, since at different points, different numbers of cycles were optimum. The solution was to select the maximum among all of the different plots and take that value to be the maximum.

Analysis	Optimum number of Cycles
Single Electron vs. QCD jets	500
Single Electron vs. $\tau \rightarrow \pi^0 \rightarrow \gamma\gamma$	1200

Table 4.2 Optimum Number of Cycles for Neural Network Analysis

For optimization of the number of nodes, similar methods to those used in the determination of cycles were used. In each case, the number of nodes was changed while each network was trained over the optimum number of cycles that was previously found. A number of different structures were attempted, with the inputs kept constant. Again, the maximum purity times signal efficiency was the quantity that was inspected, and the maximum among all of the analyses performed was taken to be the optimum number of nodes for each data set. The graphs in Figures 4.31 and 4.32

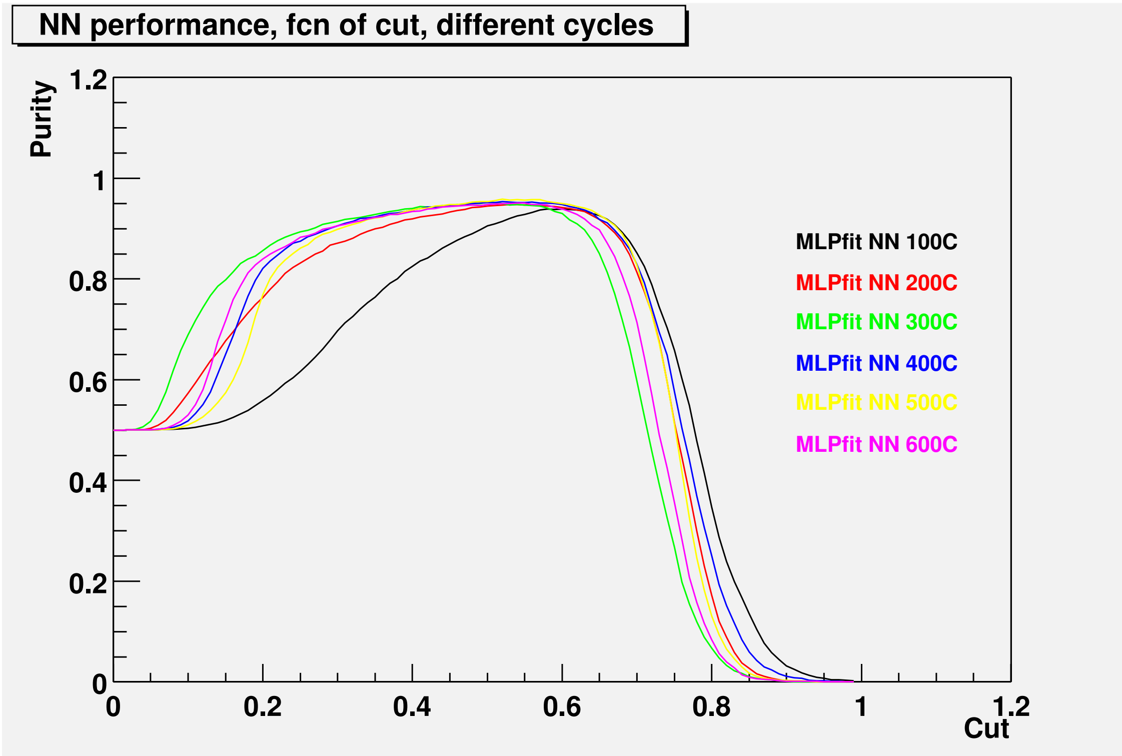

Figure 4.29 Cycles optimization: Purity times signal efficiency versus cut on neural network output for Single Electrons versus QCD jets

are the purity times signal efficiency versus a cut on the neural network output. It is noted that the number of nodes did not greatly affect the overall performance in any case, and the difference in performance at its greatest was of the order of a few percent in efficiency.

4.1.5 Optimization of Single Electron PDE Parameters

For PDE analyses, the same quantities as defined in the neural network parameters were used (efficiency, purity times signal efficiency). For the fixed kernel PDE method,

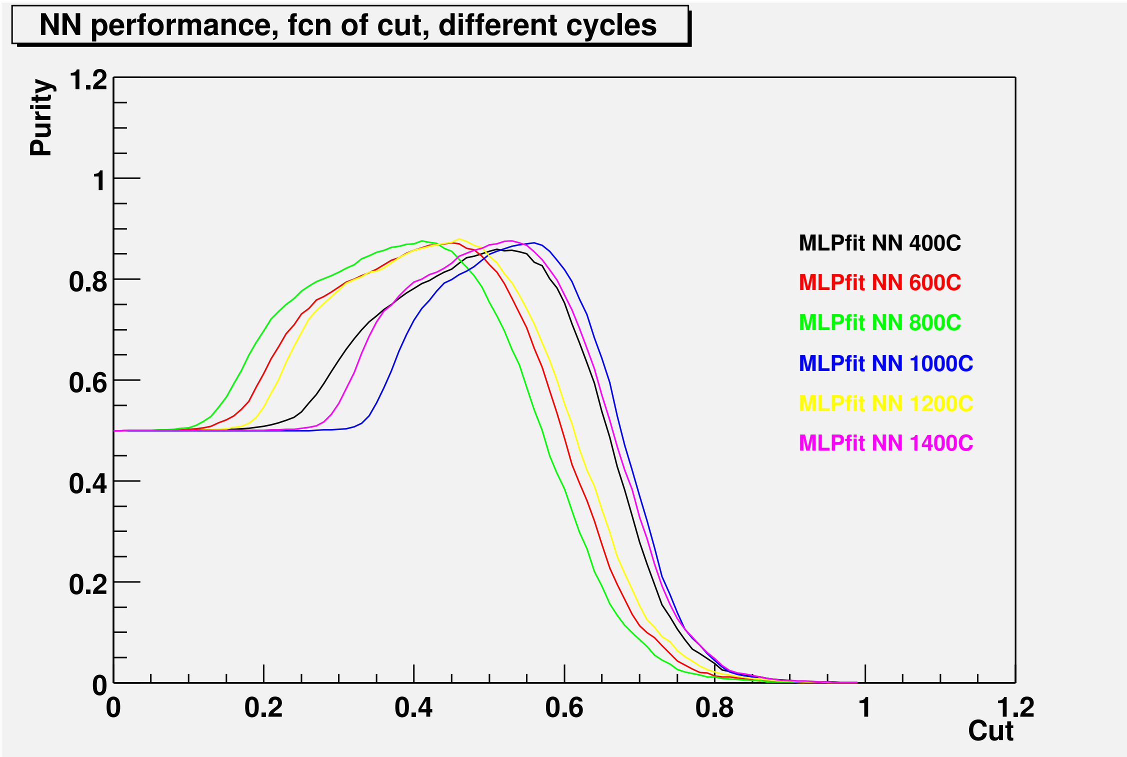

Figure 4.30 Cycles optimization: Purity times signal efficiency versus cut on neural network output for Single Electrons versus $\tau \rightarrow \pi^0 \rightarrow \gamma\gamma$

there is only one parameter to be determined for each data set, h^0. To determine the optimum value for this parameter, the analysis was performed a number of times using all of the values for h^0 between (0,1] in increments of 0.05. At each value of h^0, the purity times signal efficiency (for a discriminant value of D($\mathbf{x}$)=0.5) was computed, and then the purity times signal efficiency versus the value of h^0 was graphed. The maximum purity times signal efficiency was taken to be the optimum value of h^0 for that data set. The graphs are shown in Figures 4.33 and 4.34.

Analysis	Optimum number of Nodes
Single Electron vs. QCD jets	14
Single Electron vs. $\tau \to \pi^0 \to \gamma\gamma$	15

Table 4.3 Optimum Number of Nodes for Neural Network Analysis

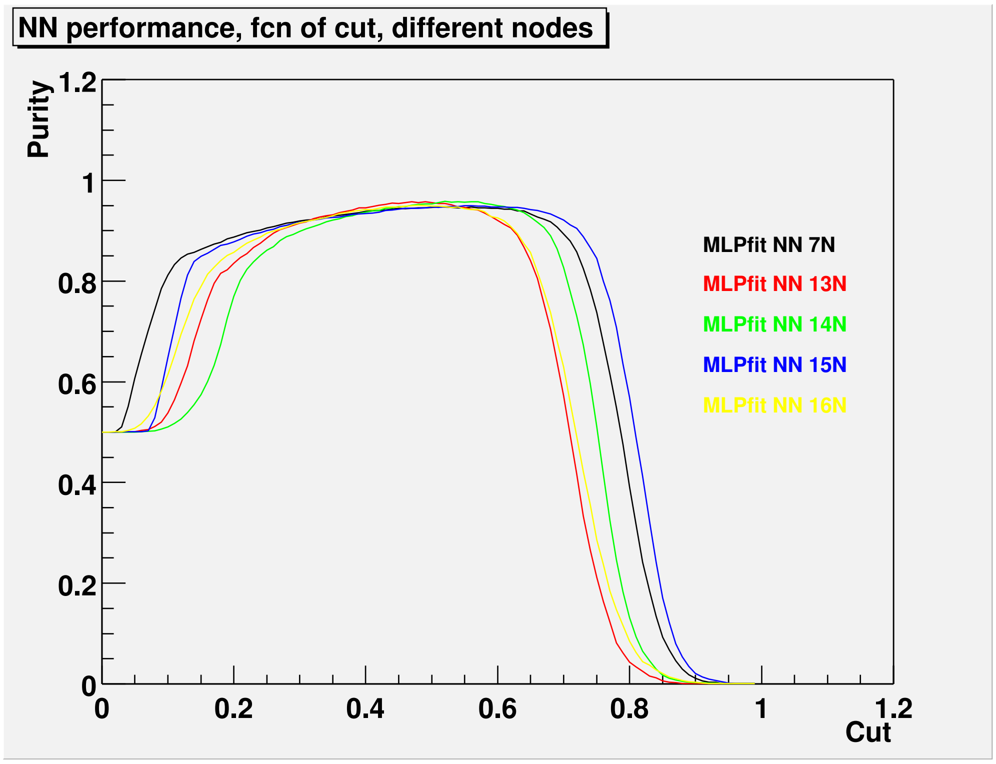

Figure 4.31 Node optimization: Purity times signal efficiency versus cut on neural network output for Single Electrons versus QCD jets

The PDE adaptive kernel has two parameters h^0 and α. The optimization of the performance for these two parameters was carried out in a similar way to the fixed kernel. In this case instead of a linear graph, a surface in the space of h^0, α and purity times signal efficiency was formed, in increments of 0.05 in h^0 and α (for the same

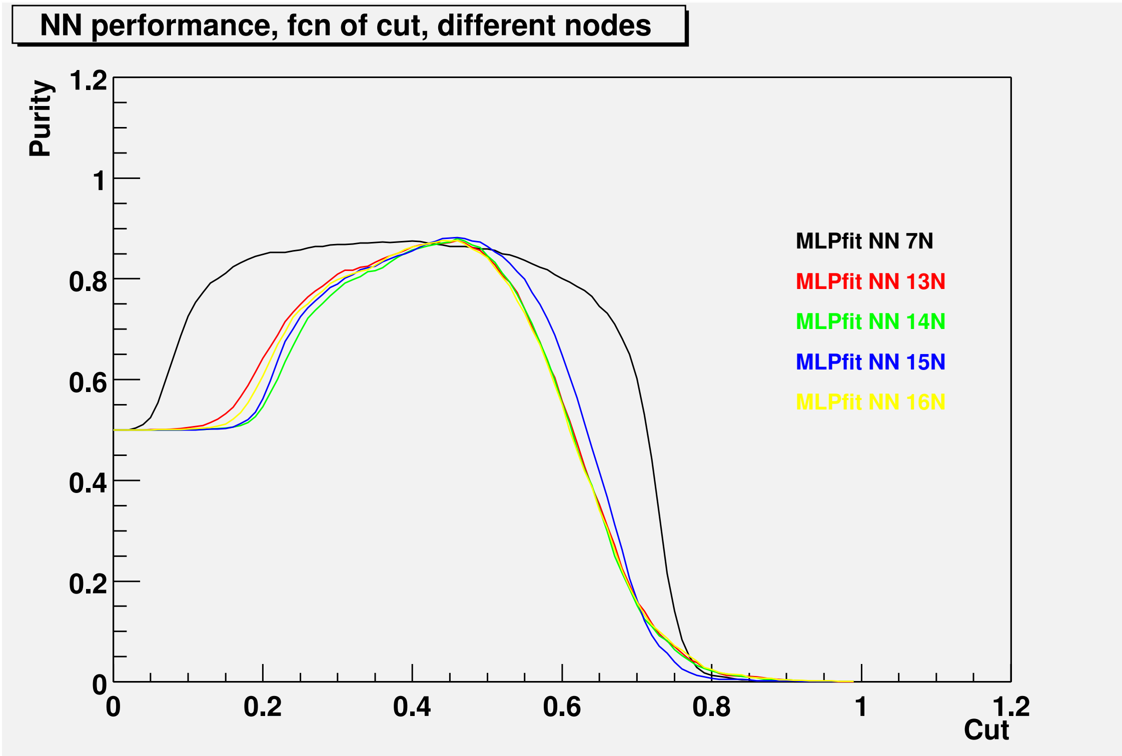

Figure 4.32 Node optimization: Purity times signal efficiency versus cut on neural network output for Single Electrons versus $\tau \to \pi^0 \to \gamma\gamma$

discriminant function value as the fixed kernel case). The maximum of this surface was taken to correspond to the optimum values of h^0 and α for the analysis.

Analysis	h^0
Single Electron vs. QCD jets	.9
Single Electron vs. $\tau \rightarrow \pi^0 \rightarrow \gamma\gamma$	.4

Table 4.4 Optimum Value of h^0 for PDE Fixed Kernel Analysis

Analysis	h^0	α
Single Electron vs. QCD jets	.5	.15
Single Electron vs. $\tau \rightarrow \pi^0 \rightarrow \gamma\gamma$	.45	.5

Table 4.5 Optimum Values of h^0 and α for PDE Adaptive Kernel Analysis

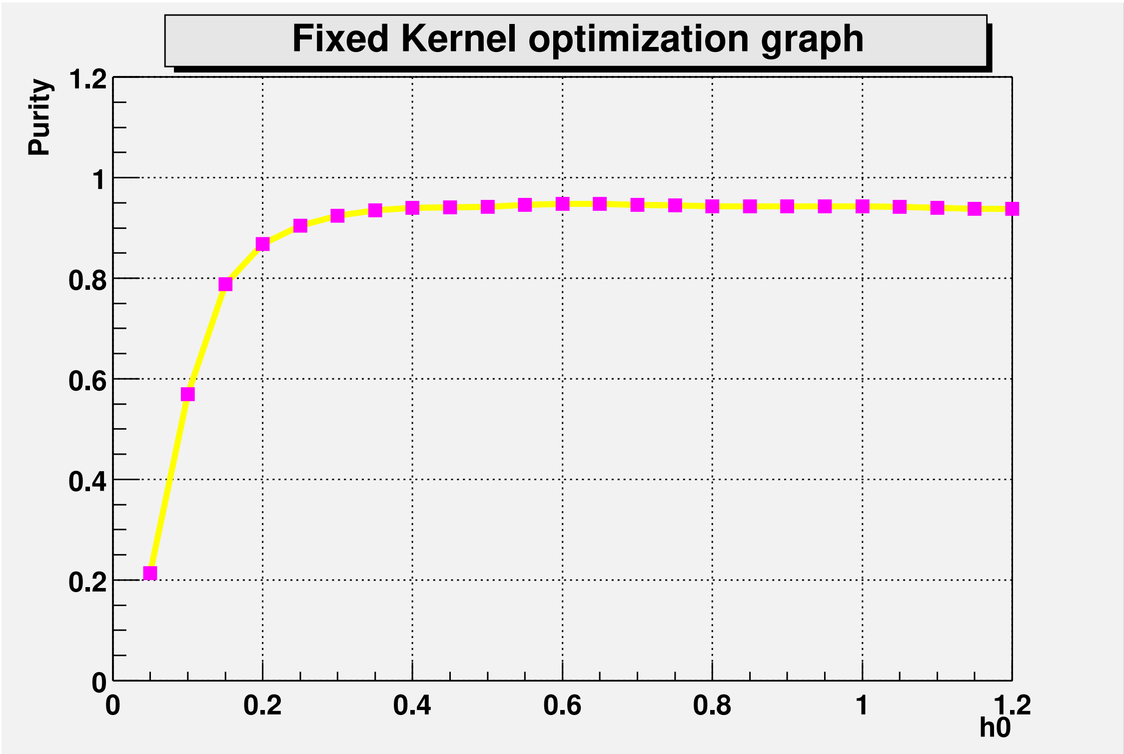

Figure 4.33 Optimization of h^0: Purity times signal efficiency versus h^0 at D = 0.5 for fixed kernel analysis of single electrons versus QCD jets

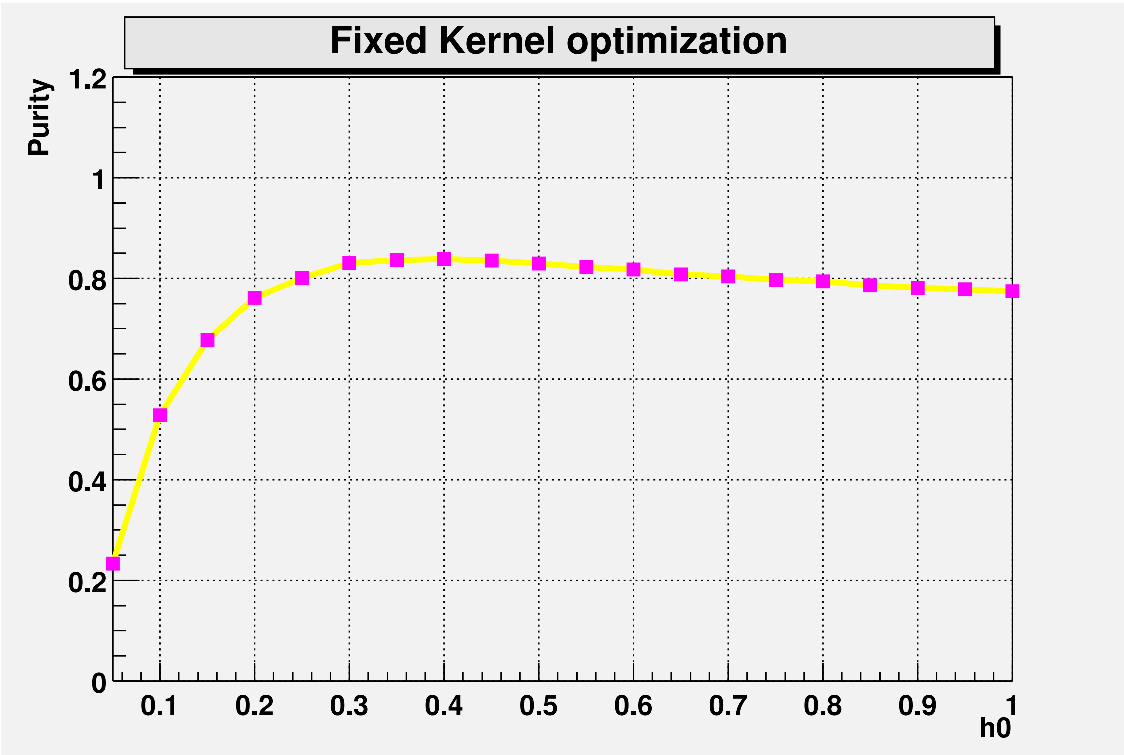

Figure 4.34 Optimization of h^0: Purity times signal efficiency versus h^0 at D = 0.5 for fixed kernel analysis of single electrons versus $\tau \rightarrow \pi^0 \rightarrow \gamma\gamma$

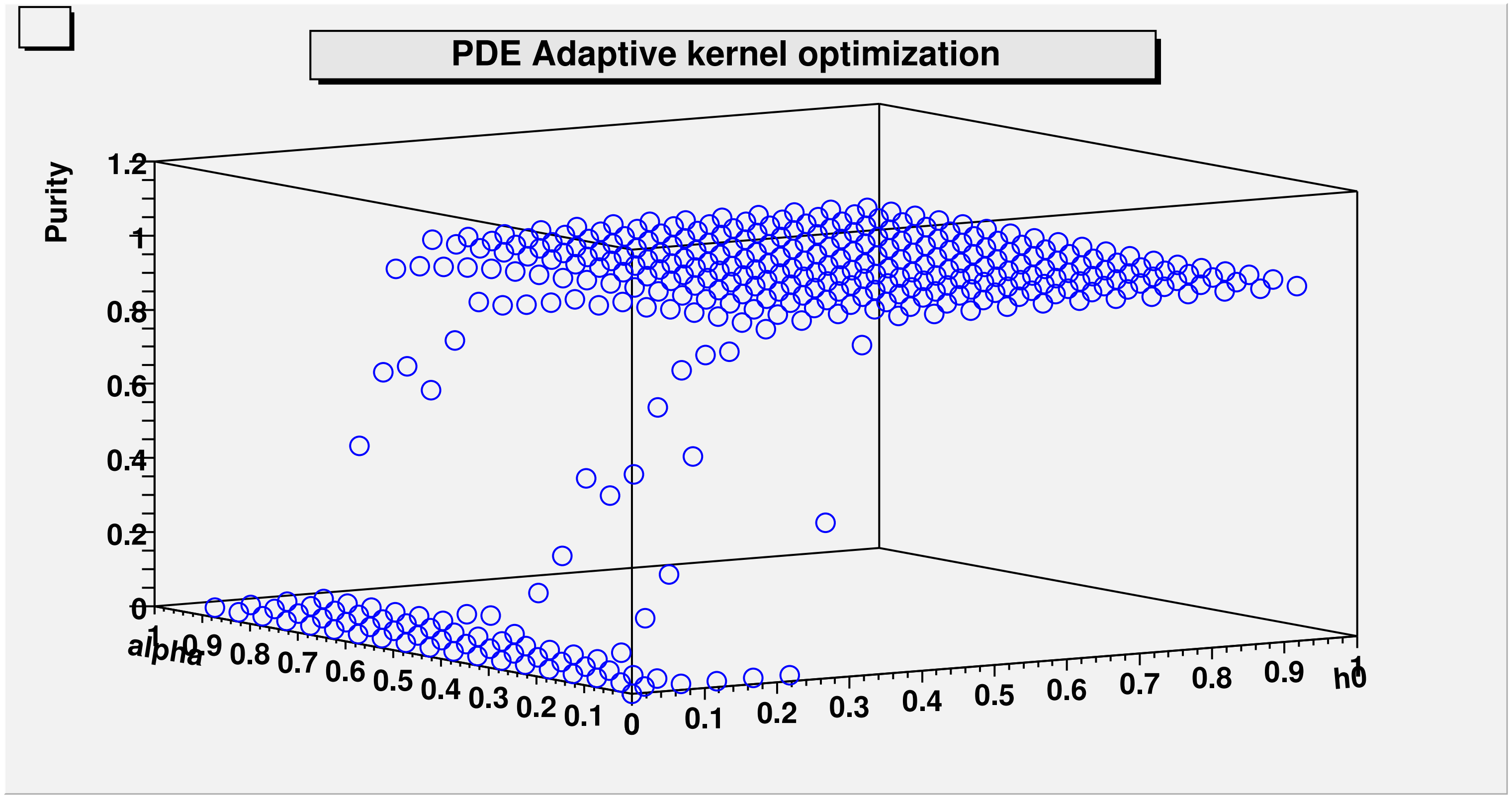

Figure 4.35 Optimization of h^0 and α: Purity times signal efficiency versus h^0 and α at D=0.5 for adaptive kernel analysis of single electrons versus QCD jets

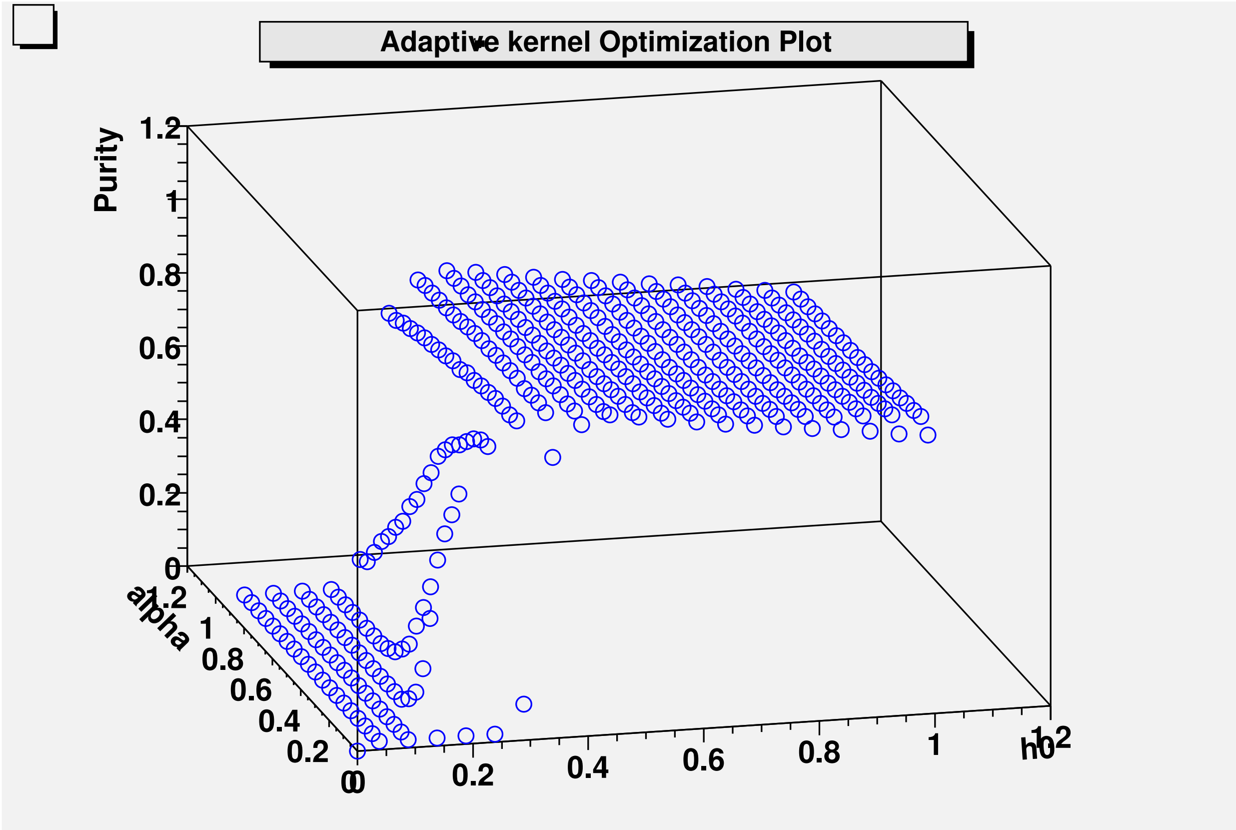

Figure 4.36 Optimization of h^0 and α: Purity times signal efficiency versus h^0 and α at D=0.5 for adaptive kernel analysis of single electrons versus $\tau \to \pi^0 \to \gamma\gamma$

4.1.6 Results

Using the optimum parameters found, the final analysis was carried out on the data sets. The graphs of Purity versus signal efficiency are shown in Figures 4.37, 4.38 and 4.39. Both cases show comparable performance between PDE and the MLPfit neural network. To find the optimal performance (the best discrimination between the signal and background), the maximum purity times signal efficiency was found for each analysis. These results are shown in Table 4.6. In these cases the new PDE adaptive kernel provides a small improvement over the fixed kernel analysis.

Method	Maximum Purity $\times\epsilon_s$	ϵ_s	ϵ_b	$\frac{\epsilon_s}{\epsilon_b}$
Single electron versus QCD jets				
Fixed Kernel PDE	.953	.982	.0299	32.8
Adaptive Kernel PDE	.957	.982	.0260	37.8
Neural Network	.958	.993	.0365	27.2
Single electron versus $\tau \to \pi^0 \to \gamma\gamma$				
Fixed Kernel PDE	.844	.935	.101	9.26
Adaptive Kernel PDE	.871	.958	.0957	10.0
Neural Network	.882	.952	.0750	12.7

Table 4.6 Best Performances of Multivariate Techniques using Maximum Purity $\times\epsilon_s$ for Single Electrons

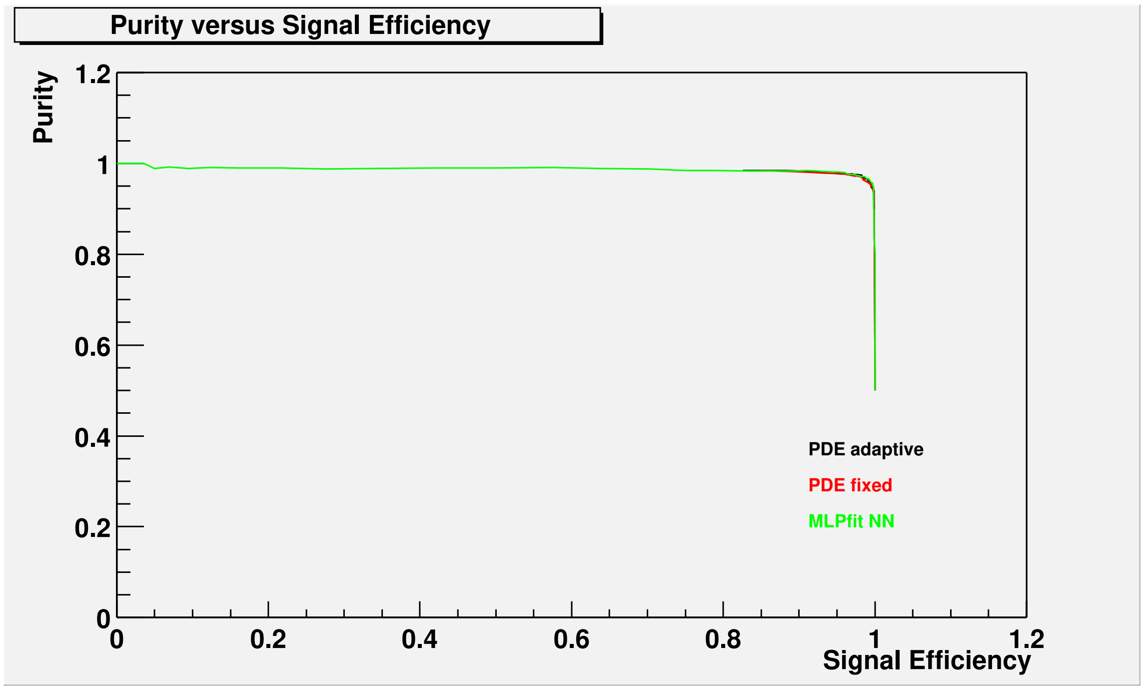

Figure 4.37 Purity versus Signal efficiency for Single Electron versus QCD jets

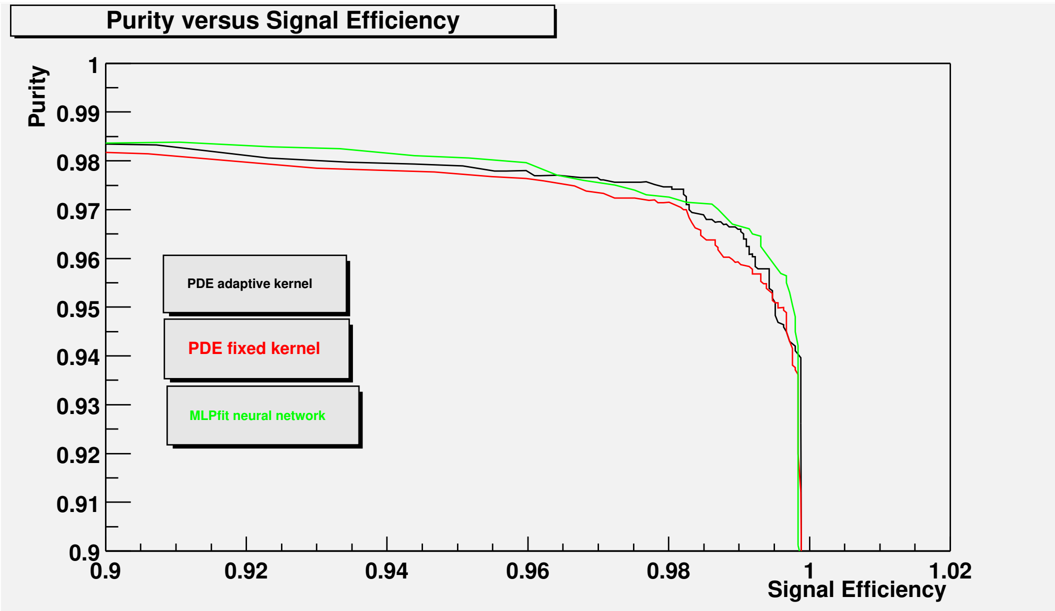

Figure 4.38 Close view of Purity versus Signal efficiency for Single Electron versus QCD jets

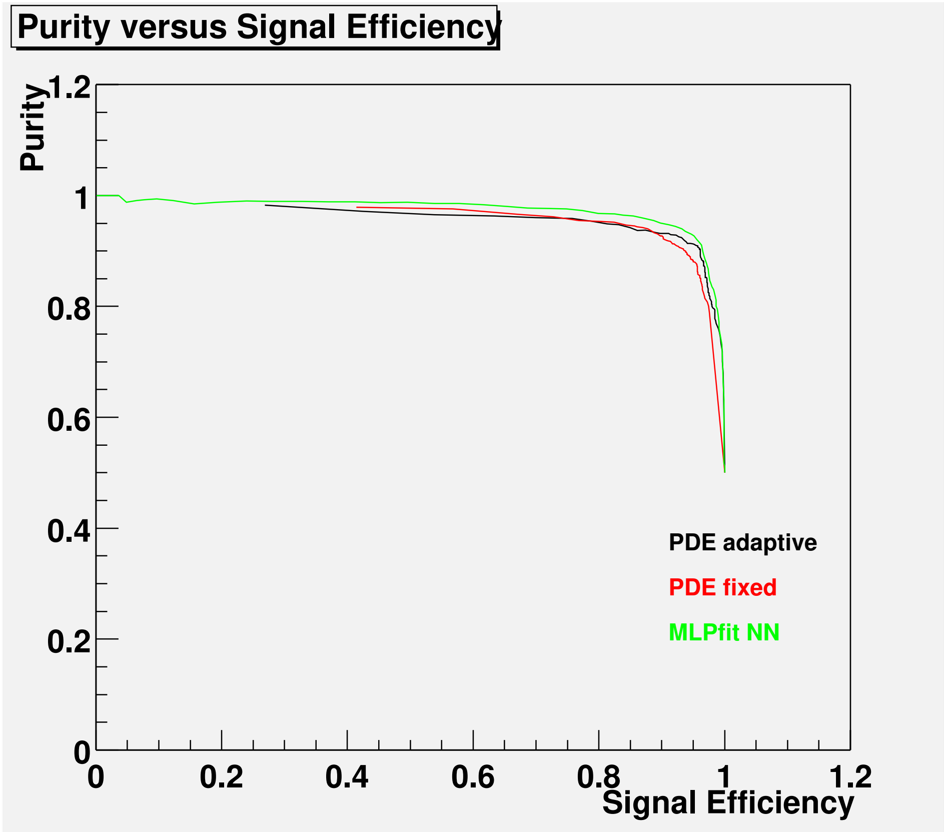

Figure 4.39 Purity versus Signal efficiency for Single Electrons versus $\tau \rightarrow \pi^0 \rightarrow \gamma\gamma$

4.2 Tau Analysis

4.2.1 Tau Identification Data Sets

The selection criteria for the Monte Carlo signal and backgrounds for the data samples selected were the same for both the training and the testing events, though as before the two data sets were independently generated. For an event to be considered, the QCD jet or hadronically decaying tau lepton was required to have the following characteristics:

1. At least one calorimeter cluster was required.

2. The cluster was required to have $E_T > 10$ GeV.

3. The cluster was required to be in the central calorimeter $|\eta| < 1.1$.

4. The cluster was required to have $p_T > 20$ GeV.

5. A track was required to be within dR<0.4 of the calorimeter cluster, where dR is defined in the definitions of the variables used for analysis below.

The signal used for training and testing data was $Z \rightarrow \tau\tau$ where the hadronic decay of only one of the tau leptons was used for the purposes of this study. QCD jets that had at least p_T >20 GeV, and met the above requirements were used as the background sample. In addition to the background, an average of 1.1 minimum bias interactions were added to all of the data sets. These interactions are additional QCD interactions in the detector, and the average indicates that these were added to the data set in a Poisson distribution with an average of 1.1. The total number of events considered, and those events surviving the cuts detailed above are summarized in Table 4.7.

	Total Number of events	Events remaining after cut
Signal Training :	1368	1065
Signal Testing :	1229	973
Background Training:	1757	1277
Background Testing :	1772	1264

Table 4.7 Events Selected for τ versus QCD jets

4.2.2 Tau Identification Variables

The following variables were used to describe the characteristics of hadronic tau decays. These represent a subset of the quantities that were originally considered during Hmatrix analysis of signal and background. The normalizations (such as the division by E_{tot}) originate there, and are kept for the purposes of comparison. These were the variables under study in neural network studies when this comparison was performed. Nine variables which include track and calorimeter information were considered, and the normalized histograms of the data sets by variable follow the definitions:

1. E_{em3}/E_{tot}: Energy deposited in the third layer of the electromagnetic calorimeter divided by the total cluster energy.

2. E_{em4}/E_{tot}: Energy deposited in the fourth layer of the electromagnetic calorimeter divided by the total cluster energy.

3. E_{fh}/E_{tot}: Energy deposited in the finely segmented layers of the hadronic calorimeter divided by the total cluster energy.

4. Crms: The calorimeter cluster RMS, defined to be

$$crms = \sqrt{\sum_i \frac{(\phi_i - \phi_c)^2 \times p_{T_i}^2}{p_T^2} + \frac{(\eta_i - \eta_c)^2 \times p_{T_i}^2}{p_T^2}}, \tag{4.7}$$

where the sum is carried out over all the calorimeter towers and: ϕ_i and η_i are the coordinates of the ith calorimeter tower, p_{T_i} is the transverse momentum measured by the ith calorimeter tower, p_T is the total transverse momentum measured in the calorimeter, and ϕ_c, η_c are the coordinates of the calorimeter cluster.

5. E_{hot2}/E_{tot}: The sum of the energy deposited in the two hottest towers (the two towers with the greatest energy deposition) in the calorimeter divided by the total cluster energy. This is also sometimes referred to as the Profile.

6. $\log_{10} E_{tot} - 1$: Logarithm (base 10) of the total cluster energy minus one.

7. Trms: The track RMS, defined to be

$$trms = \sqrt{\sum_i \frac{(\phi_i - \phi_0)^2 \times p_{T_i}^2}{p_T^2} + \frac{(\eta_i - \eta_0)^2 \times p_{T_i}^2}{p_T^2}}, \tag{4.8}$$

where the sum is carried out over all of the tracks (within dR<0.4 of the cluster) and (ϕ_i, η_i) are the coordinates of each individual track, with p_{T_i} being the tracks transverse momentum, p_T is the sum of all the track transverse momenta (tracks within dR<0.4) and,

$$\phi_0 = \frac{\sum_i(\phi_i p_{T_i})}{\sum_j(p_{T_j})}, \eta_0 = \frac{\sum_i(\eta_i p_{T_i})}{\sum_j(p_{T_j})}, \tag{4.9}$$

where these individual sums are over the calorimeter towers, with (η_j, ϕ_j) being the tower's coordinates, p_{T_j} the measured transverse momentum in each tower.

8. $\ln Nt_2$: Natural logarithm of the number of tracks within a dR<0.4 of the calorimeter cluster where dR is defined to be

$$dR = \sqrt{(\phi - \phi_C)^2 + (\eta - \eta_C)^2}, \tag{4.10}$$

where (η, ϕ) are the coordinates of the track and (η_C, ϕ_C) are the coordinates of the calorimeter cluster.

9. $\frac{Nt_2 - Nt_1}{Nt_2 + 1}$: Number of tracks within dR<0.4 of the calorimeter cluster minus the number of tracks within dR<0.2 of the cluster, divided by the number of tracks within dR<0.4 plus one (for normalization), where dR is as defined above. This provides information on the track isolation fraction.

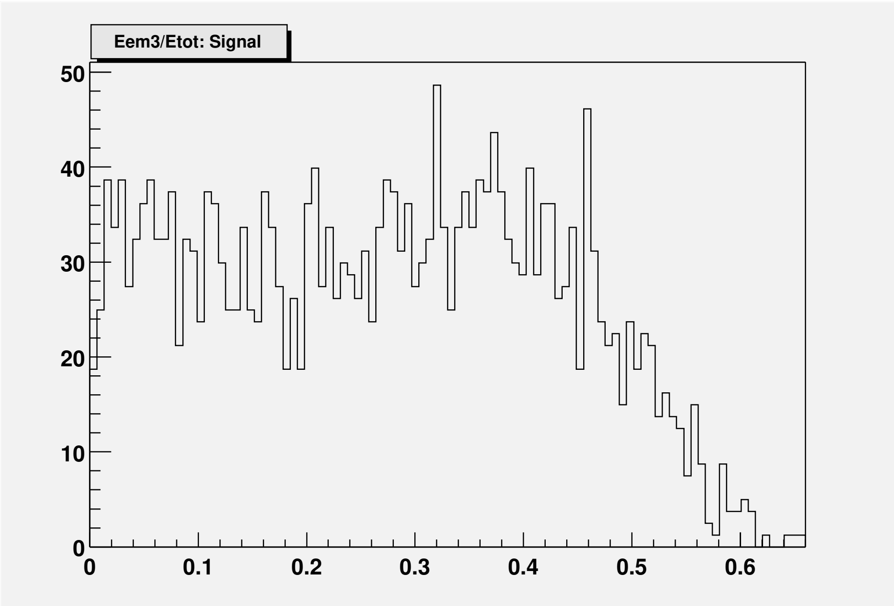

Figure 4.40 E_{em3}/E_{tot} for signal

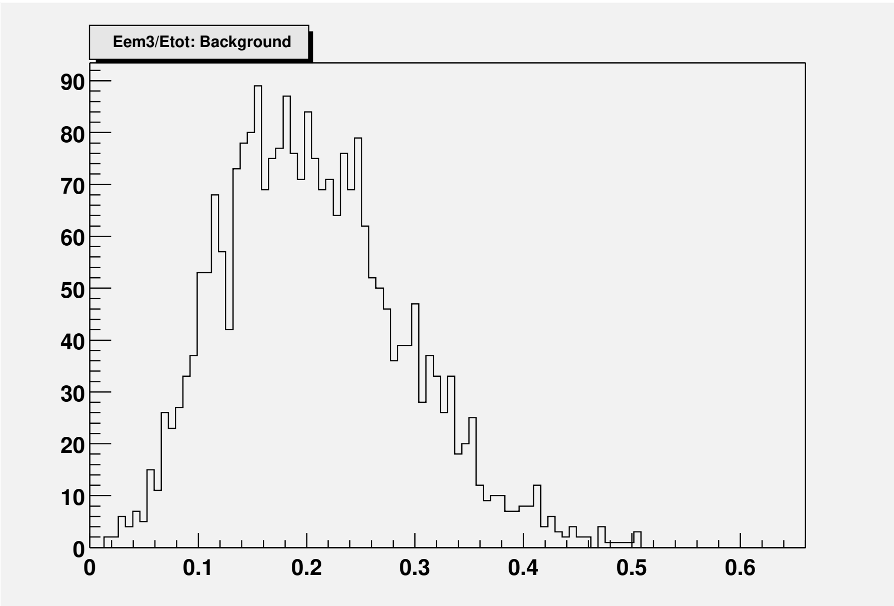

Figure 4.41 E_{em3}/E_{tot} for background

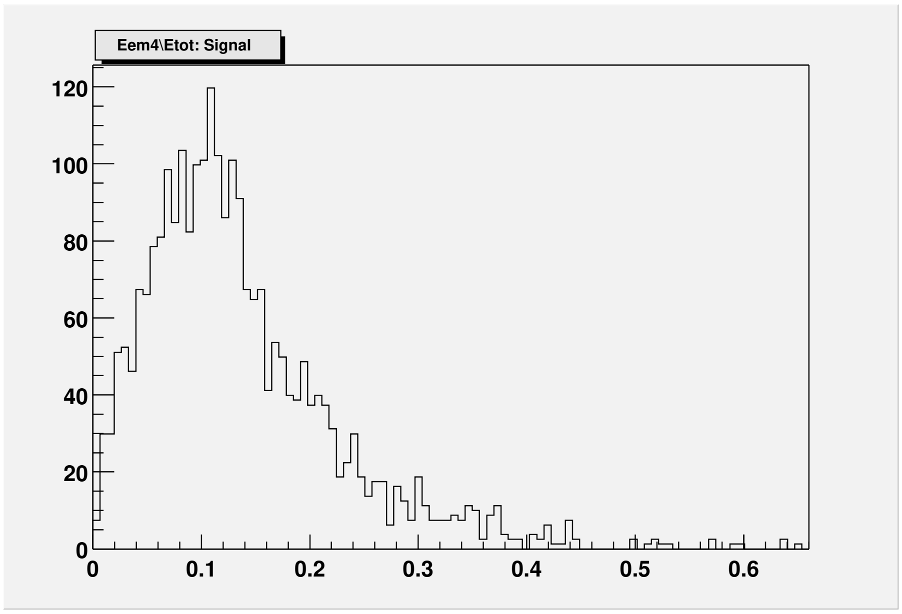

Figure 4.42 E_{em4}/E_{tot} for signal

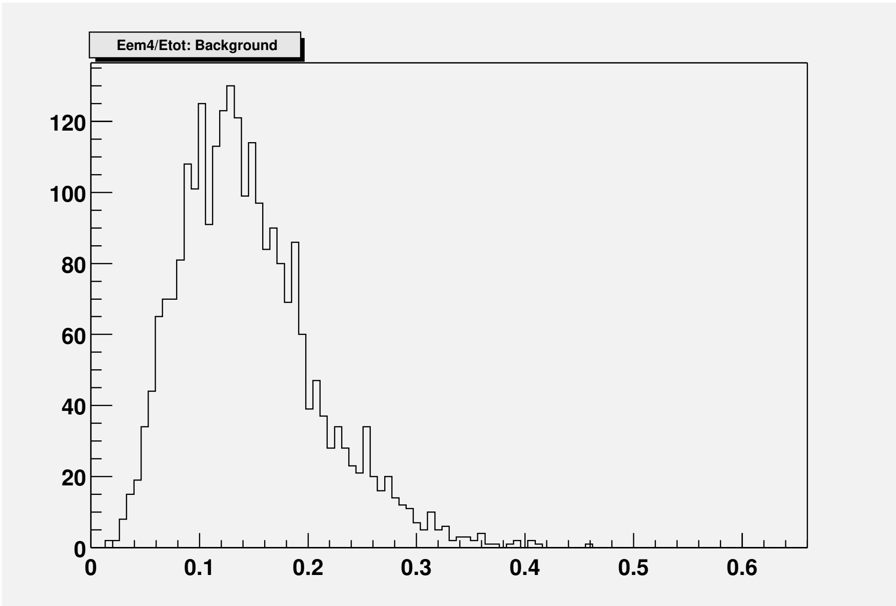

Figure 4.43 E_{em4}/E_{tot} for background

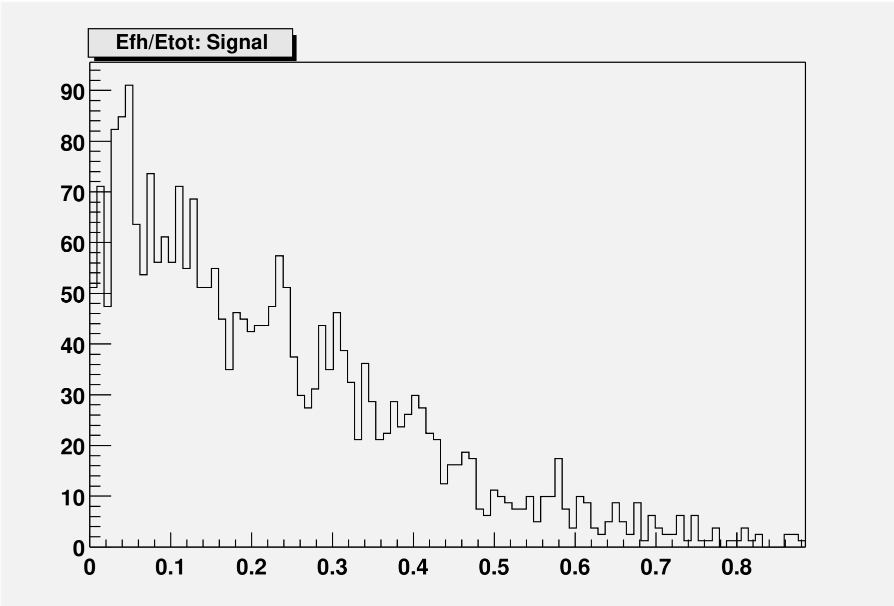

Figure 4.44 E_{fh}/E_{tot} for signal

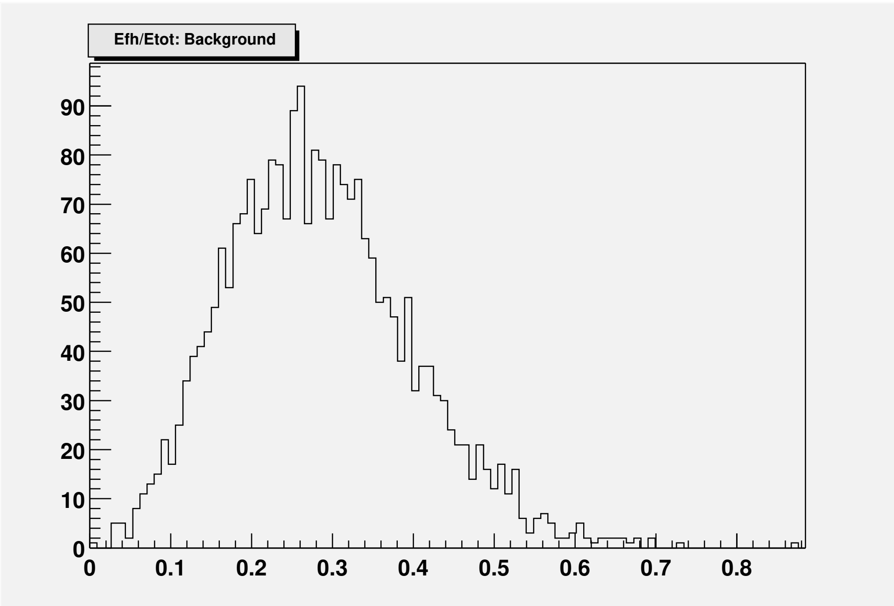

Figure 4.45 E_{fh}/E_{tot} for background

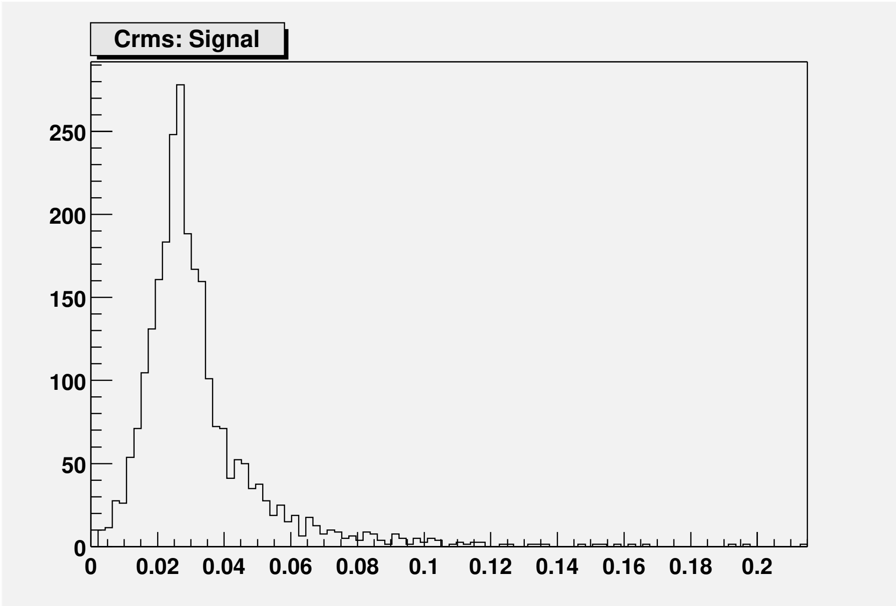

Figure 4.46 Crms for signal

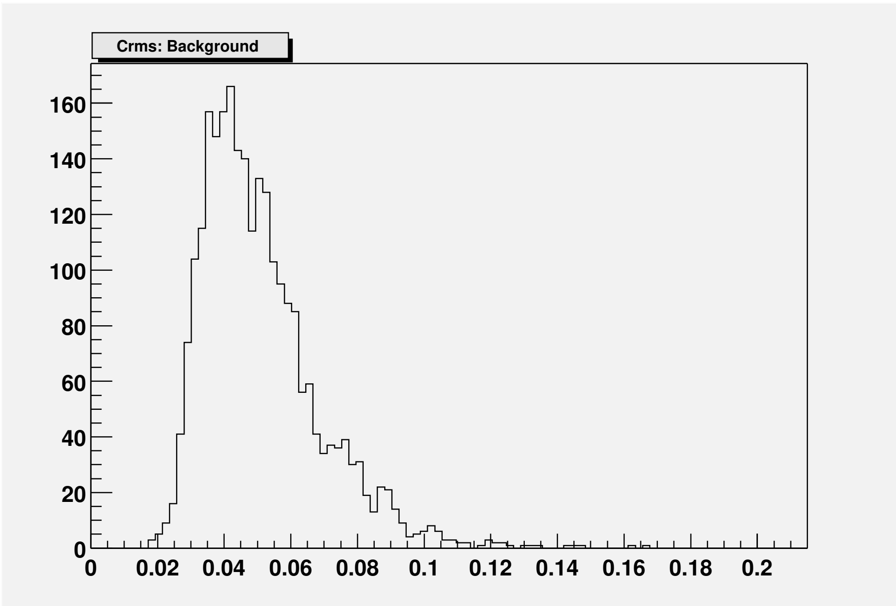

Figure 4.47 Crms for background

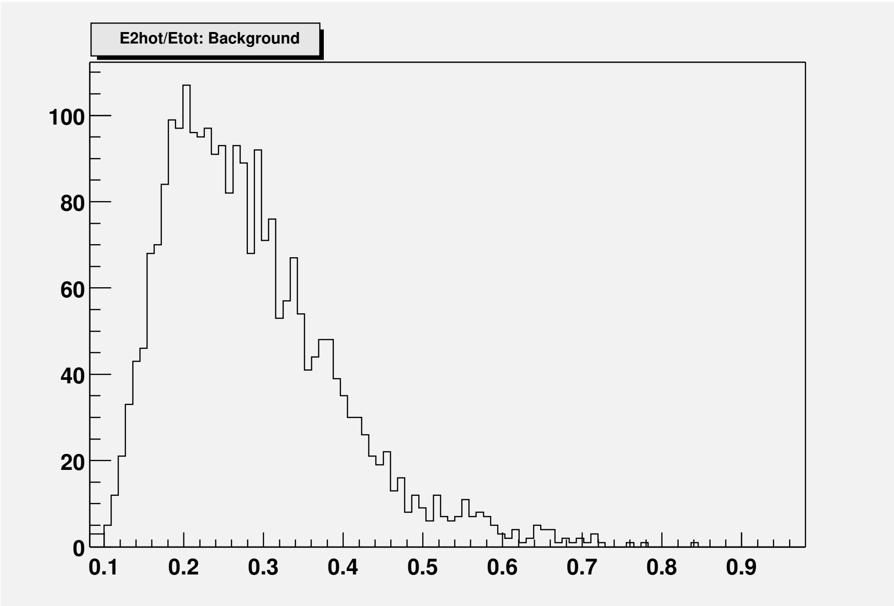

Figure 4.48 E_{hot2}/E_{tot} for signal

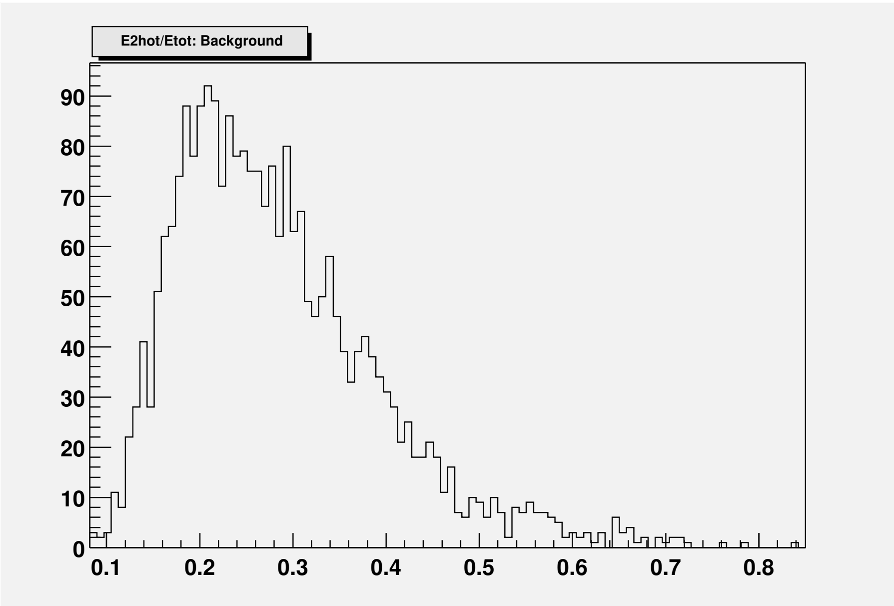

Figure 4.49 E_{hot2}/E_{tot} for background

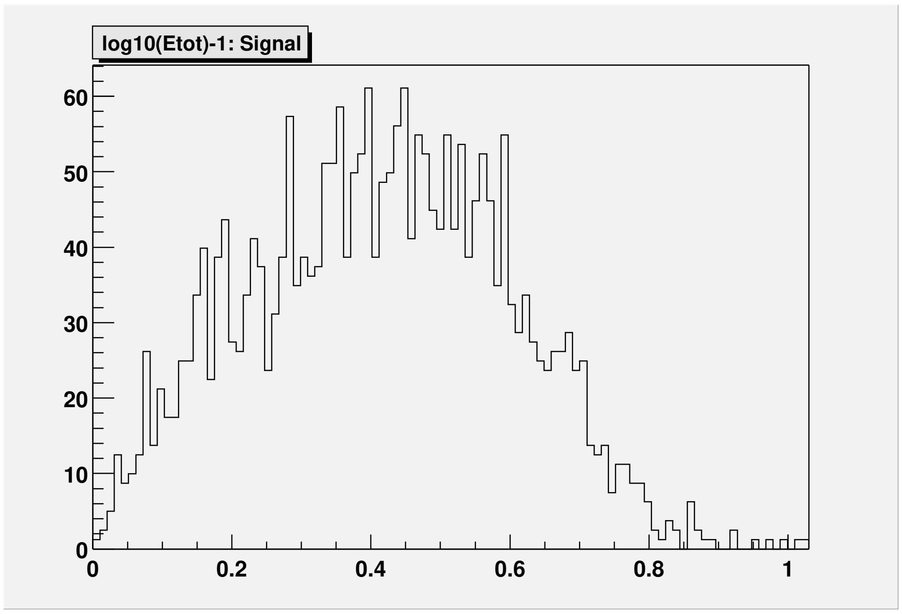

Figure 4.50 $\log_{10} E_{tot} - 1$ for signal

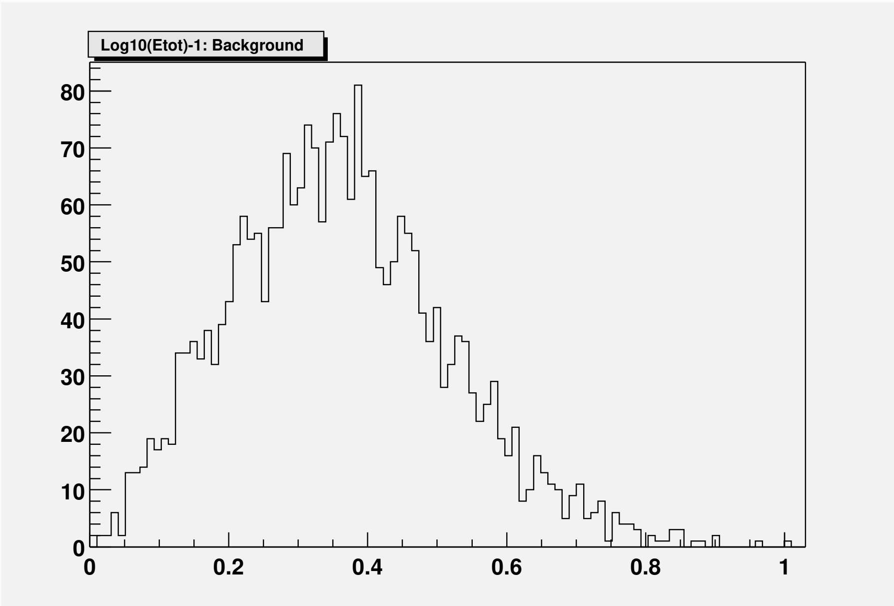

Figure 4.51 $\log_{10} E_{tot} - 1$ for background

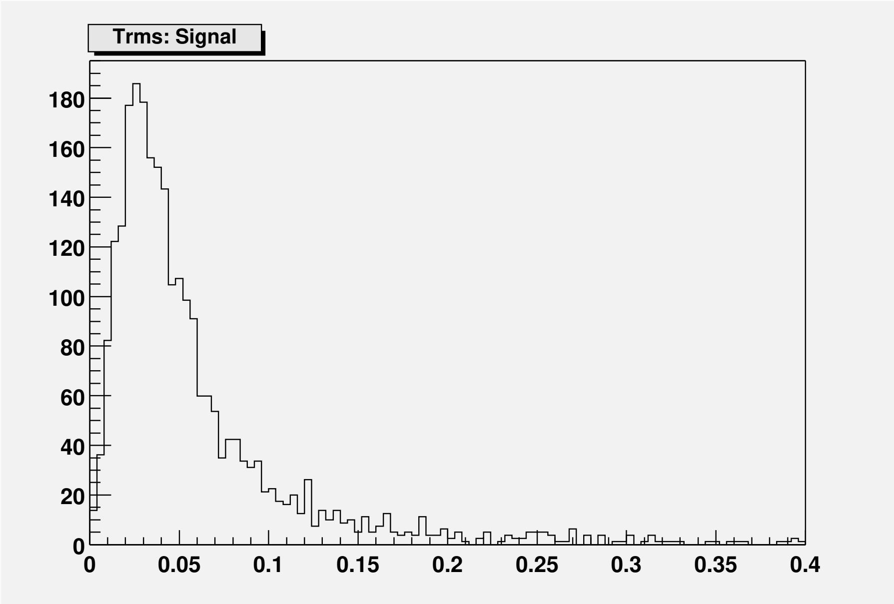

Figure 4.52 Trms for signal

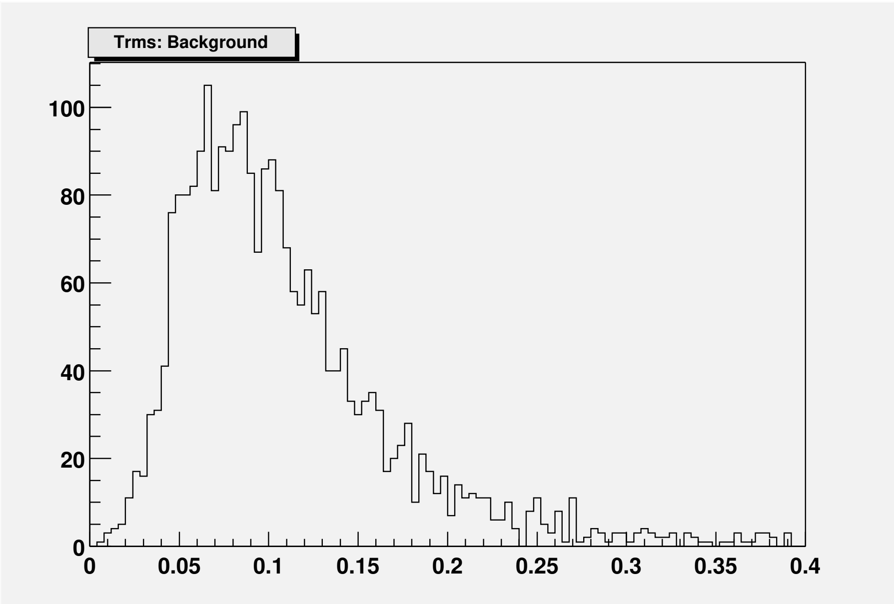

Figure 4.53 Trms for background

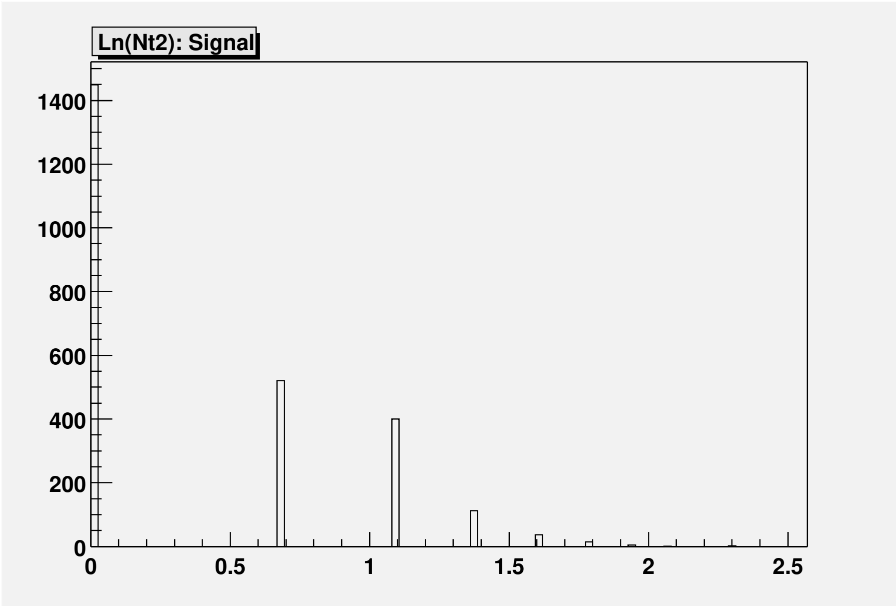

Figure 4.54 $\ln Nt_2$ for signal

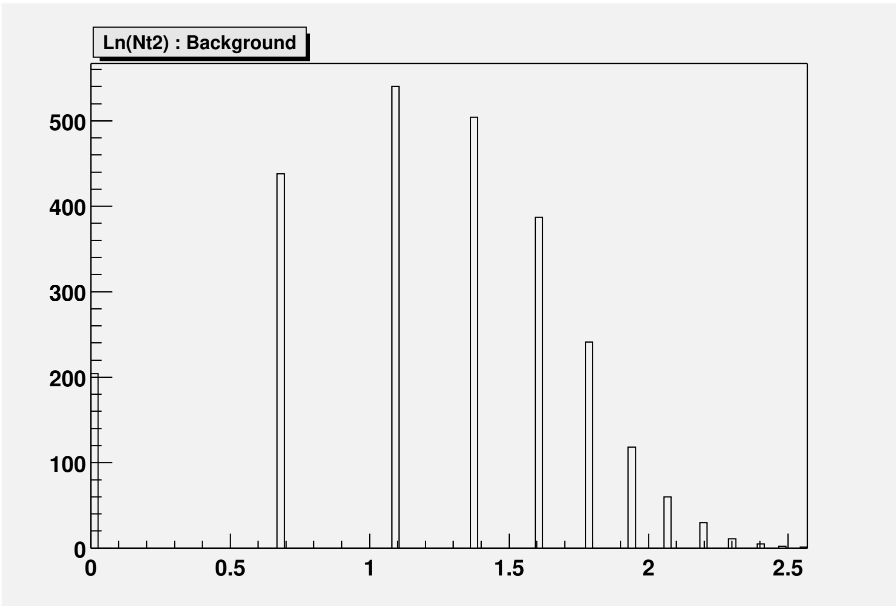

Figure 4.55 $\ln Nt_2$ for background

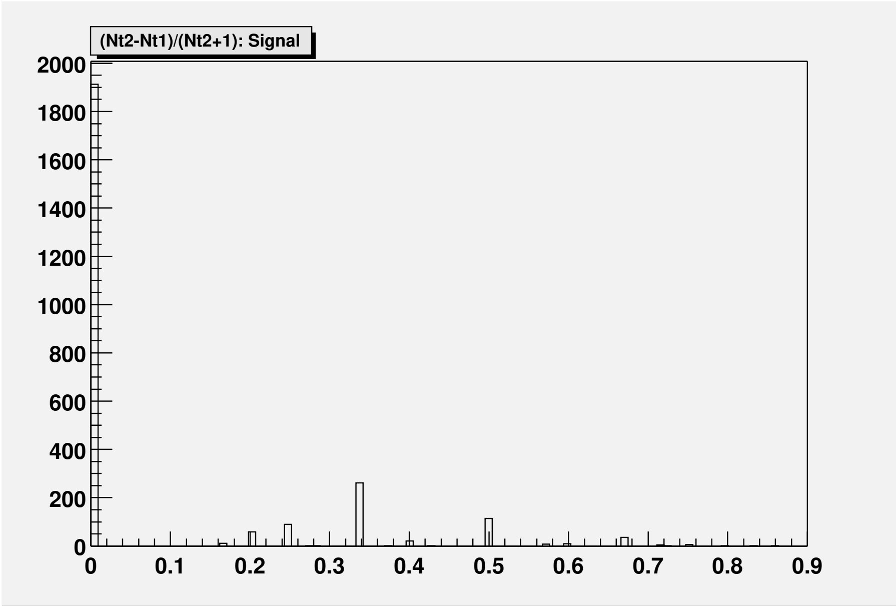

Figure 4.56 $\frac{Nt_2-Nt_1}{Nt_2+1}$ for signal

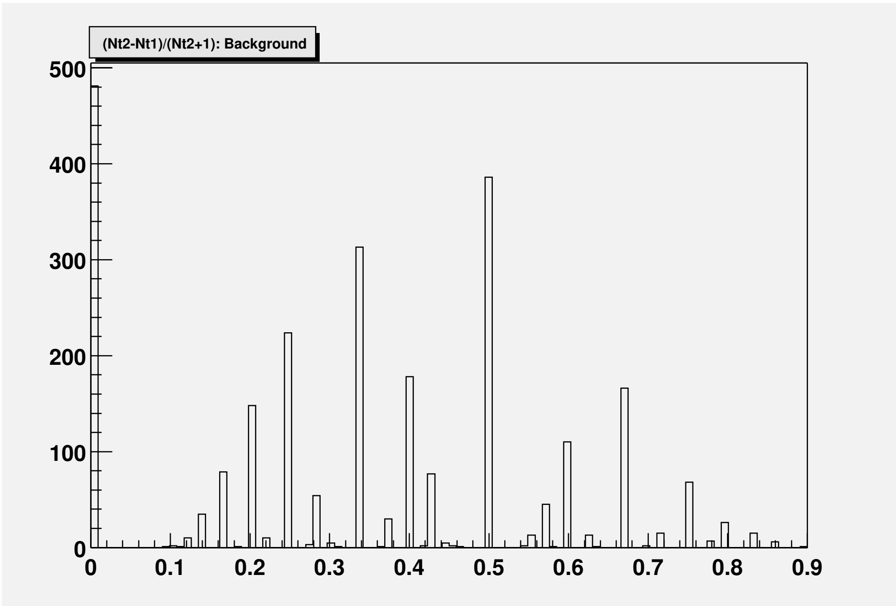

Figure 4.57 $\frac{Nt_2-Nt_1}{Nt_2+1}$ for background

4.2.3 Optimization of Tau Neural Network Parameters

The optimization of the neural network for the identification of tau leptons was done in the same manner as described for the single electron analyses. Here the result was that the optimum number of cycles was 200 and the optimum number of nodes was twenty-two (where the optimum value corresponds to the maximum signal efficiency times purity for the different networks trained).

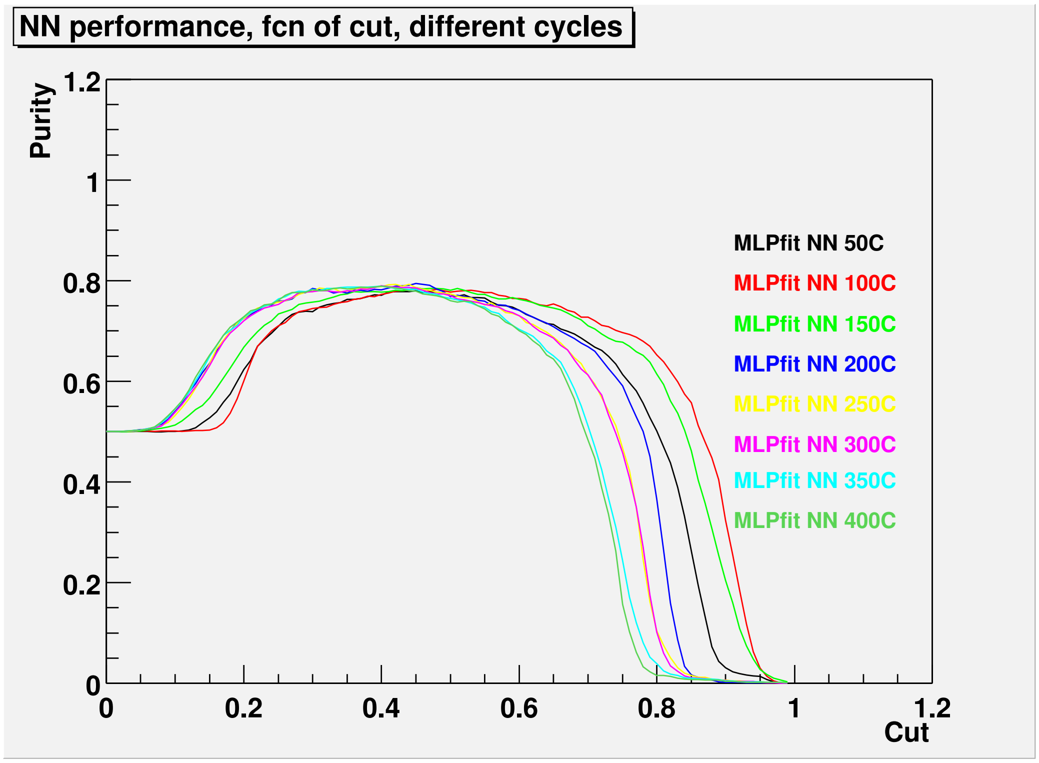

Figure 4.58 Cycles optimization: Purity times signal efficiency versus cut on neural network output for Tau leptons versus QCD jets

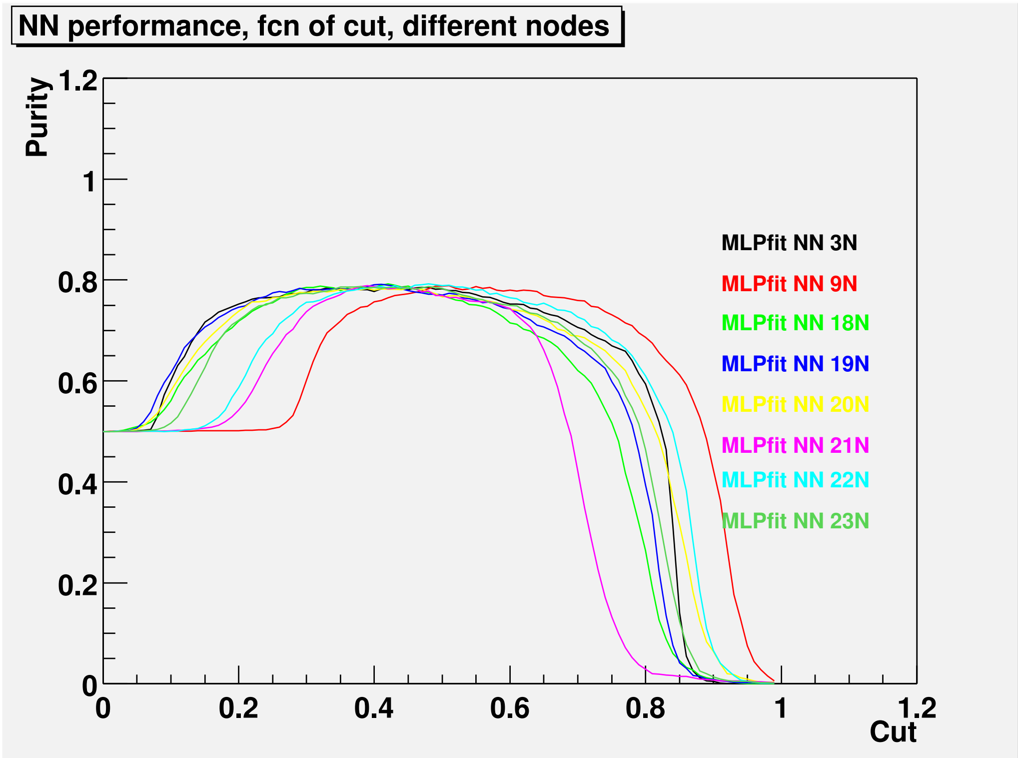

Figure 4.59 Node optimization: Purity times signal efficiency versus cut on neural network output for Tau leptons versus QCD jets

4.2.4 Optimization of Tau PDE parameters

The optimization performed for both the fixed kernel and the adaptive kernel PDE analyses on tau lepton data sets was performed as described in the single electron case. In the fixed kernel case an optimum value of $h^0 = 0.55$ was obtained. In the adaptive kernel analysis, the maximum of the surface was found to be at $(h^0 = 0.65, \alpha = 0.25)$. The graphs for these optimizations are shown in Figures 4.60 and 4.2.4.

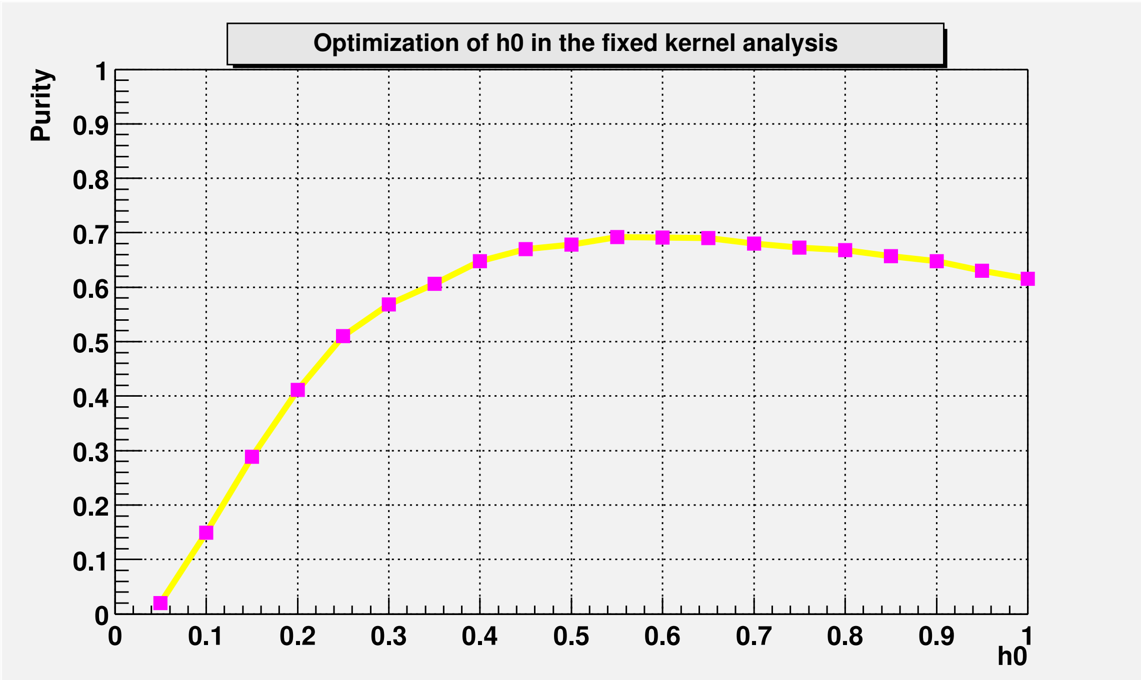

Figure 4.60 Optimization of h^0: Purity times signal efficiency versus h^0 at D = 0.5 for fixed kernel analysis of τs versus QCD jets

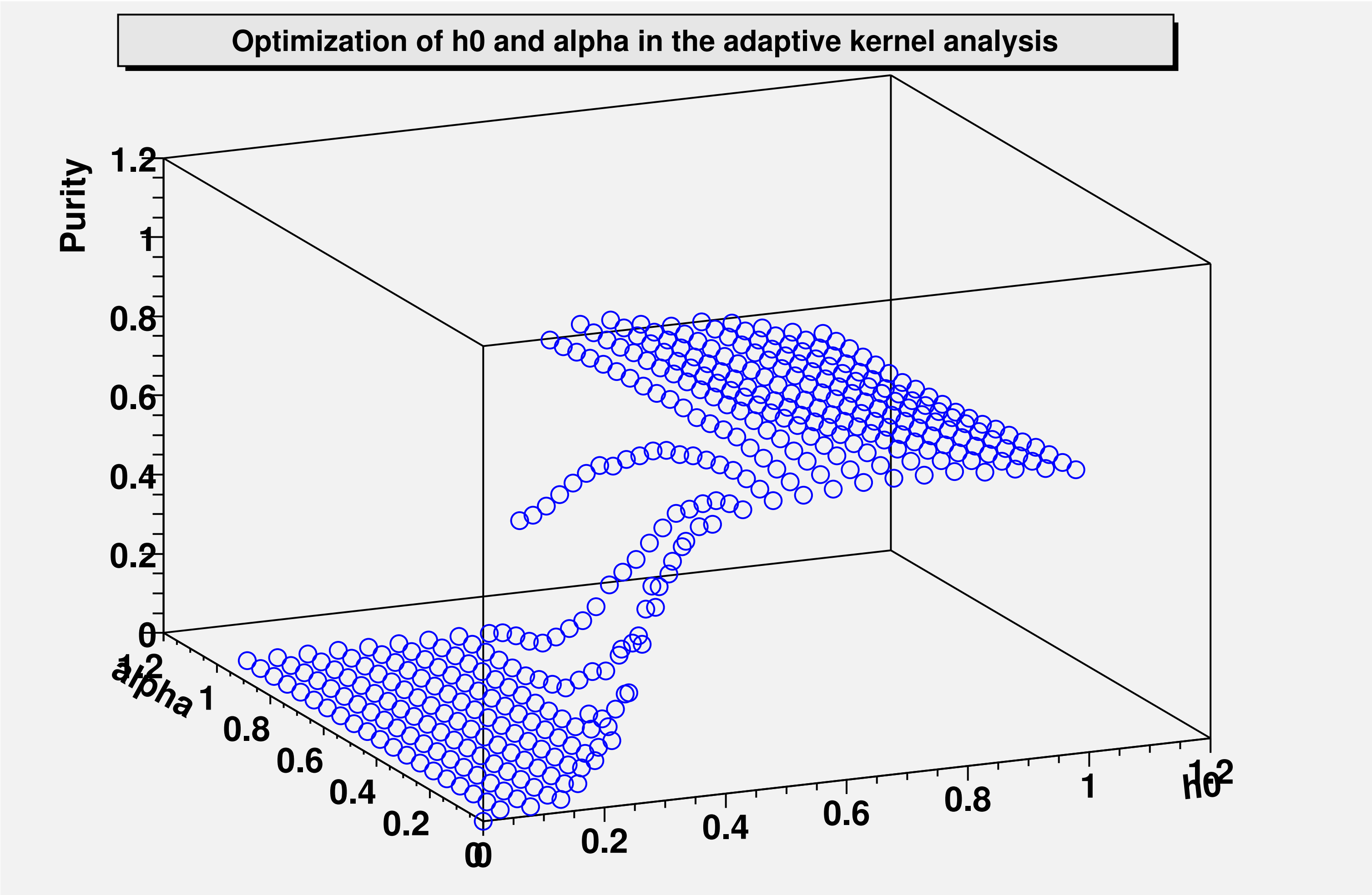

Figure 4.61 Optimization of h^0 and α: Purity times signal efficiency versus h^0 and α at D=0.5 for adaptive kernel analysis of τs versus QCD jets

4.2.5 Results

Using the optimum parameters, the analyses were performed on the tau lepton data set. The Purity versus signal efficiency is shown in Figure 4.62 for all three methods. In this case, the adaptive kernel PDE provides a more noticeable improvement over the fixed kernel analysis. The optimum performance (best discrimination between signal and background) was chosen to be the maximum purity times signal efficiency. These performances are listed in Table 4.8. The multivariate techniques show comparable performance in the identification of tau leptons.

Method	Maximum Purity $\times\epsilon_s$	ϵ_s	ϵ_b	$\frac{\epsilon_s}{\epsilon_b}$
τ versus QCD jets				
Fixed Kernel PDE	.731	.879	.178	4.94
Adaptive Kernel PDE	.781	.861	.0886	9.72
Neural Network	.793	.883	.100	8.83

Table 4.8 Best Performances of Multivariate Techniques using Maximum Purity $\times\epsilon_s$ for Tau Leptons

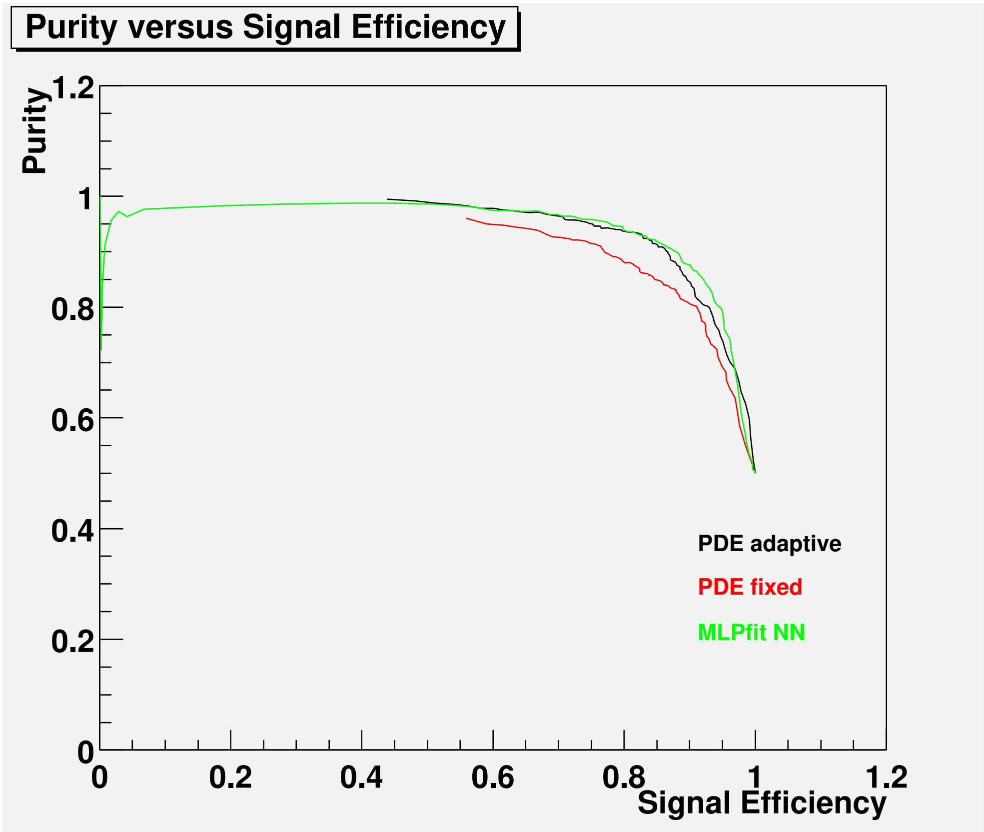

Figure 4.62 Signal versus Background efficiency for τ versus QCD jets

Chapter 5
Conclusion

5.1 Summary

The performance of probability density estimation and neural networks have been shown in the previous chapters. Both give comparable efficiencies for discriminating between the signal events and the background events. These results are summarized in Table 5.1. Though these two multivariate techniques give almost the same efficiencies, these results still represent an improvement over Hmatrix studies and linear cuts. These results show a definite improvement over recent studies done using linear cuts and Hmatrix in the identification of tau leptons.

Method	Maximum Purity $\times\epsilon_s$	ϵ_s	ϵ_b	$\frac{\epsilon_s}{\epsilon_b}$
τ versus QCD jets				
Fixed Kernel PDE	.731	.879	.178	4.94
Adaptive Kernel PDE	.781	.861	.0886	9.72
Neural Network	.793	.883	.100	8.83
Single electron versus QCD jets				
Fixed Kernel PDE	.953	.982	.0299	32.8
Adaptive Kernel PDE	.957	.982	.0260	37.8
Neural Network	.958	.993	.0365	27.2
Single electron versus $\tau \to \pi^0 \to \gamma\gamma$				
Fixed Kernel PDE	.844	.935	.101	9.26
Adaptive Kernel PDE	.871	.958	.0957	10.0
Neural Network	.882	.952	.0750	12.7

Table 5.1 Summary of Multivariate Performances

5.2 Future Prospects

The use of probability density estimation and neural networks show great promise as methods by which particles may be identified. Further studies carried out should concentrate on the number and type of variables chosen to characterize the data sets. It is conceivable that with fewer variables the discrimination may improve. Also, for discrete variables such as the number of tracks that are matched with the calorimeter clusters, the multivariate techniques examined here may be at a disadvantage since the input space for both neural networks and PDE is taken to be a continuous distribution. Different data sets that include more minimum bias events, more closely simulating the conditions of Run II need to be studied.

Appendix A
Neural Network Weight Optimization

The central computational problem of neural networks is the optimization of the weights. There are many algorithms that can be used, but for the case of O(10) inputs and O(100) hidden nodes, a quasi-Newton method is the least computationally intensive [6]. The simplest way of regarding the problem of minimizing the error function is to treat it as a function of a vector of weights ω, which is a W-dimensional vector with the adaptive parameters (weights and offsets) of the neural network as its components. Then vector algebra techniques can be used in the minimization. The Newton method of weight optimization is fairly straightforward, and will be briefly described.

Taking the truncated Taylor expansion of the error function, expressing the output as a function of a vector of weights,

$$E(\omega) = E(\hat{\omega}) + (\omega - \hat{\omega})^T \times b + (\omega - \hat{\omega})^T H(\omega - \hat{\omega}), \tag{A.1}$$

the local expansion is;

$$\nabla E \approx b + H(\omega - \hat{\omega}), \tag{A.2}$$

where both of the above are for near $\hat{\omega}$, H is the Hessian matrix ($\frac{\partial^2 E}{\partial \omega_i \partial \omega_j}$), and b is the gradient of the error function evaluated at $\hat{\omega}$. If ω^* is the weight vector that minimizes the error function then the expansion about this minimum is (since b=0 at ω^*);

$$\nabla E = g = H(\omega - \omega^*) \tag{A.3}$$

and therefore,

$$\omega^* = \omega - H^{-1} \times g. \tag{A.4}$$

The quantity $\mathrm{H}^{-1} \times \mathrm{g}$ is known as the Newton direction and points toward the vector in weight space that minimizes the error function. Therefore, to optimize the weights for the neural network, all that must be done is to move in this direction in weight space until the minimum error is achieved. Calculation of the Hessian matrix and then its inverse is computationally expensive. Therefore, an algorithm that provides successive approximations to the inverse Hessian matrix is utilized (thus these methods are termed Quasi-Newton). The Broyden-Fletcher-Goldfarb-Shanno (BFGS) expression is one such iterative approximation, given by the matrix G;

$$G^{(\tau+1)} = G^{(\tau)} + \frac{pp^T}{p^T\nu} - \frac{(G^{(\tau)}\nu)\nu^T G^{(\tau)}}{\nu^T G^{(\tau)}\nu} + (\nu^T G^{(\tau)}\nu)uu^T, \tag{A.5}$$

where;

$$p = \omega^{(\tau+1)} - \omega^{(\tau)}, \tag{A.6}$$

$$\nu = g^{(\tau+1)} - g^{(\tau)}, \tag{A.7}$$

$$u = \frac{p}{p^T\nu} - \frac{G^{(\tau)}\nu}{\nu^T G^{(\tau)}\nu}, \tag{A.8}$$

where $\omega^{(\tau)}$ is the τth vector of weights, $\mathrm{G}^{(\tau)}$ is the τth approximation of the inverse Hessian matrix, $\mathrm{g}^{(\tau)}$ is the τth gradient of the error function (evaluated at ω^τ), and the superscript T indicates that the transpose is being taken, as these are vector and matrix equations. When this method is used, G is initially the identity matrix ($\tau = 0$), and the starting vector of weights is chosen by a random number generator (between -.5 and .5) to avoid bias in the starting point. Then the first error function is computed using these random weights. A new vector of weights is chosen via a line search along the direction of $G^{(\tau)}g^{(\tau)}$, in other words, choosing the point along this vector for which the error no longer decreases. $G^{(\tau+1)}$ is then computed using the new $\omega^{(\tau+1)}$ found by this line minimization and the process repeats. Each computation of G is considered one training epoch. This continues until the output of the error function is considered to have reached its minimum. The termination of this process (number of iterations performed) can be considered another parameter to be tuned on the data set (others being number of nodes and the number and types of inputs used). The BFGS method is the algorithm implemented in the Physics Analysis Workstation (PAW) package, which was used in the comparison.

Appendix B
PDE analysis script

```
{//A. Askew 7-31-00
//Seperated PDE analysis macro
//using MC (Pythia) ntuples
//This file does a 9 variable analysis, both adapative and
//fixed kernal on Z->tau tau signal vs. QCD background.

TFile f1("normpythiatrainsig.root");
TTree *t1=(TTree*)f1.Get("h3");        //Training signal tree

TFile f2("normpythiatrainback.root");
TTree *t2=(TTree*)f2.Get("h3");        //Training background tree

TFile f3("normpythiatestsig.root");
TTree *t3=(TTree*)f3.Get("h3");        //Testing signal tree

TFile f4("normpythiatestback.root");
TTree *t4=(TTree*)f4.Get("h3");        //Testing background tree
gROOT.cd();

gSystem->Load("pde.so");             //Load the pde object
gStyle->SetOptStat(000000);

pde p;                                 //declare a pde object

p.SetTrees(t1,t2);   //Set the trees that will create the signal
                     //and background functions.  The first is
                     //signal, the second background.

TList *list = new TList;
TString sx = "e3norm";
TString sy = "e2hotnorm";
TString sz = "e4norm";
TString sa = "efhnorm";         //Declare a list of the variables
TString sc = "etotnorm";        //int the trees that the analysis
TString sd = "lnt2";            //is to be performed over.
```

```
TString se = "nt2mnt1";
TString sf = "ncrms";
TString sg = "ntrms";
list->Add((TObject*)\&sx);
list->Add((TObject*)\&sy);
list->Add((TObject*)\&sz);
list->Add((TObject*)\&sa);
list->Add((TObject*)\&sc);
list->Add((TObject*)\&sd);
list->Add((TObject*)\&se);
list->Add((TObject*)\&sf);
list->Add((TObject*)\&sg);

p.SetGlobals(0.65,0.25);  //Set h0 and alpha, the two PDE adaptive
                          //parameters.  The first value is h0, the
                          //second is alpha.  For fixed kernel, the
                          //second parameter is ignored, but a value
                          //must be supplied.

p.BuildArrays(list);      //Read the values from the trees into the
                          //PDE data objects for analysis.

p.DoAdaptive();           //Perform the analysis.  Create fs and fb.

TTree DASignalTree;
DASignalTree = p.DoTest(t3);  //Using the above functions, creates
                              //similar functions for the data input.
                              //In this case, a linearly independent
                              //set of signal data.

TTree DABackTree;
DABackTree = p.DoTest(t4);    //In this case background.

pde g;                        //Create a new pde object.

g.SetTrees(t1,t2);

g.SetGlobals(0.55,0);         //Same procedure as before...

g.BuildArrays(list);
```

```
g.Do();                       //Here, the adaptive kernel will not
                              //be used.  Instead a fixed kernel PDE
                              //analysis will be performed.

TTree DNSignalTree;
DNSignalTree = g.DoTest(t3);
                             //Here we get the results from our test.
TTree DNBackTree;
DNBackTree = g.DoTest(t4);
                             //Write the trees with our completed
                             //analysis out to files:
TFile PDEresa("tauPDEAdSigresult.root","RECREATE");
DASignalTree.Write();
TFile PDEresb("tauPDEAdBackresult.root","RECREATE");
DABackTree.Write();
TFile PDEresc("tauPDEFixSigresult.root","RECREATE");
DNSignalTree.Write();
TFile PDEresd("tauPDEFixBackresult.root","RECREATE");
DNBackTree.Write();
}
```

References

1. D. Chakraborty, “Tau Identification Using Neural Networks”
(URL: `http://www-d0.fnal.gov/~d0upgrad/d0_private/software/tauid/vreview/tau_nn.ps`)

2. “The DØ Detector (“The NIM paper”)”, FERMILAB-Pub-93/179-E (1994).

3. “The DØ Upgrade: The Detector and Its Physics”, FERMILAB-Pub-96-357-E (1996).

4. “The DZERO Fiber Tracker Technical Design Report”,
(URL:
`http://d0server1.fnal.gov/useres/stephan/www/CFT_TDR/CFT_TDR.ps`)

5. C. Bishop, *Neural Networks for Pattern Recognition.* Claredon Press (1995).

6. J, Schwindling, B. Mansoulie, O. Couet. “Using Multi-Layer Perceptrons in PAW”,
(URL: `http://wwwinfo.cern.ch/asd/paw/mlpfit/pawmlp.html`)

7. H. Miettinen, R. Ou, L. Holmstrom, S. Sain, “ Searching for Top with Neural Nets II: NN versus Probability Density Estimation”, DØ Internal Note 1931 (1993), unpublished.

8. J. Gentle, *Numerical Linear Algebra for Applications in Statistics.* Springer-Verlag (1998).