Decision Tree Technique for Particle Identification

Ryan Quiller

Office of Science, SULI Program

Rensselaer Polytechnic Institute

Stanford Linear Accelerator Center

Menlo Park, California

September 5, 2003

Prepared in partial fulfillment of the requirements of the Office of Science, DOE Student Undergraduate Laboratory Internship (SULI) Program under the direction of Dr. Charles C. Young in Experimental Group A at the Stanford Linear Accelerator Center.

Participant: _____

Signature

Research Advisor: _____

Signature

Table of Contents

Abstract	3
Introduction	4
Materials and Methods	7
Results	10
Discussion and Conclusions	11
Acknowledgements	13
References	13
Tables	14
Figures	17

Abstract

Decision Tree Technique for Particle Identification. RYAN QUILLER (Rensselaer Polytechnic Institute, Troy, New York, 12180) CHARLES C. YOUNG (Stanford Linear Accelerator Center, Menlo Park, California 94025).

Particle identification based on measurements such as the Cerenkov angle, momentum, and the rate of energy loss per unit distance ($-dE/dx$) is fundamental to the BaBar detector for particle physics experiments. It is particularly important to separate the charged forms of kaons and pions. Currently, the Neural Net, an algorithm based on mapping input variables to an output variable using hidden variables as intermediaries, is one of the primary tools used for identification. In this study, a decision tree classification technique implemented in the computer program, CART, was investigated and compared to the Neural Net over the range of momenta, 0.25 GeV/c to 5.0 GeV/c. For a given subinterval of momentum, three decision trees were made using different sets of input variables. The sensitivity and specificity were calculated for varying kaon acceptance thresholds. This data was used to plot Receiver Operating Characteristic curves (ROC curves) to compare the performance of the classification methods. Also, input variables used in constructing the decision trees were analyzed. It was found that the Neural Net was a significant contributor to decision trees using dE/dx and the Cerenkov angle as inputs. Furthermore, the Neural Net had poorer performance than the decision tree technique, but tended to improve decision tree performance when used as an input variable. These results suggest that the decision tree technique using Neural Net input may possibly increase accuracy of particle identification in BaBar.

Introduction

Particle identification based on the BaBar detector's data is crucial for much of the research conducted at the Stanford Linear Accelerator Center. In order to accurately determine the particles being produced in the detector and their decay products, it is necessary to have a systematic way of distinguishing particles. Among the particles that are detected in BaBar, it is important to be able to make a highly efficient distinction between the charged forms of pions and kaons.

These mesons can be separated if their momenta and velocities are known. The momentum of a charged particle can be determined from the curvature of its path in the magnetic field of the detector. The velocity of a particle can be deduced from a few sources including data from the silicon vertex tracker (SVT), drift chamber, and the Detector of Internally Reflected Cerenkov radiation (DIRC). The SVT and drift chamber provide measurements of the mean rate of energy loss per unit distance, $-dE/dx$, which will characterize the particle's speed. However, as shown in Figure 1, at approximately 1.0 GeV/c, pions and kaons are expected to have the same mean rate of energy loss, which creates a problem in differentiating particles [1]. Likewise, DIRC provides data related to the Cerenkov cone half-angle, θ_c , which distinguishes the particle's speed. This is shown in Figure 2, and it can be noted that for increasing momentum, the separation between the pion and kaon curves diminishes, making it harder to distinguish the particles. Also, since all of these measurements are subject to experimental uncertainties, particle identification is not a trivial process.

Currently, several particle identification techniques are being implemented, such as the Neural Network. The Neural Network is an algorithm that learns to recognize patterns in data so

it can map input variables to an output, using hidden layers as intermediate steps [2]. The focus of this research is an alternative to the Neural Network, the decision tree technique, implemented in a computer program, CART.

The fundamental idea of the decision tree technique is to use a learning dataset to determine binary splits of the data into smaller subsets, or nodes, so that each descendent subset is purer than its parent subset in some class, such as pions or kaons. To find the binary set of questions used for splitting, there must be a goodness of split criterion that can be evaluated for any split of any node. This criterion is related to the decrease in impurity associated with a given split. Therefore, the split giving the largest decrease in impurity for a given node can be found. There must also be a rule to stop the binary splits, resulting in final subsets of data known as terminal nodes. The splitting stops when a node is reached where no significant decrease in impurity is possible. Once the tree is developed, there should be a process for assigning every terminal node to a class so that the tree may be used as a predictive tool for other sets of data. This assignment is usually based on which class has the greatest probability in the terminal node. Consequently, if a profile is dropped down the tree and it ends in a terminal node that has been labeled class x , then the profile will be labeled class x [3].

As mentioned above, in developing these techniques for particle identification, it is necessary to begin with a dataset in which the particle type for each case is already known. For example, in CART, this dataset would be the learning dataset, which is used to form a tree that would optimize the trade-off between accuracy and simplicity in separating pion and kaon cases. After developing and testing the tree, it can be used to classify particles in other datasets where the particles have not yet been identified.

For this research, the dataset that will be used for creating and testing decision trees in CART is from the decay products of D^{*+} and D^{*-} . This sample is used because its allowed decay processes imply what its daughter particles must be to high precision. More specifically, it is known that D^{*+} decays to $\pi^+ D^0$ with a 67.7% branching ratio and that D^0 decays to $K^- \pi^+$ with a branching ratio of 3.80% and to $K^+ \pi^-$ with the branching ratio, 0.0148% [1]. This means D^0 decays to $K^- \pi^+$ instead of $K^+ \pi^-$ approximately 250 times more often. Therefore, examining the final products of this D^{*+} decay process, it is clear that the two particles of the same charge must be pions and the particle of opposite charge must be a kaon. Also, the pion produced in the D^{*+} decay to $\pi^+ D^0$ has characteristically low momentum, which allows separation of these events from the background due to random combinations of particles. Similar conclusions can be deduced from D^{*-} decay. Since this information separates pions and kaons, each case in the dataset can be labeled. Clearly, this provides useful learning and testing datasets for developing decision trees.

The purpose of this classification study is to compare the performance of decision trees with the performance of the Neural Network. Also, this work investigates the potential for decision trees to supplement the Neural Network in increasing its predictive power in identifying particles. Based on the decision trees developed in CART, this study will determine the importance of the measured variables at specific momenta to check the trees' predictive structures. Furthermore, this work will provide information that may be used to create more efficient particle identification methods, which will improve the accuracy of particle physics experiments.

Materials and Methods

The pion and kaon dataset used in this study contained 358,421 cases. A random subset of this was used as the learning sample for the Neural Net, and output from the Neural Net was then included with each case in the entire dataset as possible input variables for use in decision trees. Using this pion and kaon dataset, two random subsets were chosen as the learning and testing sets for the decision trees created in CART. These subsets contained 119,478 and 119,057 cases, respectively. The testing subset used with the decision tree was used to analyze the performance of both the decision tree and Neural Net techniques.

Before developing decision trees and analyzing classification performance, the two subsets of data were further restricted in order to reduce misidentification. For example, cuts were imposed on the variable corresponding to the DIRC acceptance angle. This limited the data to cases with the angle between 0.449718 radians and 2.54863 radians. Likewise, cuts were imposed on the variables corresponding to the mass of D^0 and the difference in mass between D^0 and D^{*+} . The mass of D^0 was restricted to the range of values between 1.8445 GeV/c^2 and 1.8845 GeV/c^2 . The difference in mass between D^0 and D^{*+} was confined to the interval, 0.1445 GeV/c^2 to 0.1465 GeV/c^2 . These two cuts reduced the number of cases associated with random combinations of particles. The distribution functions of the three variables used to impose cuts are shown in Figure 3, 4, and 5. After the cuts were imposed, the resulting learning and testing subsets of data contained 93,129 and 92,842 cases, respectively.

To check which variables provide the best separation of pions and kaons and to study the performance of decision trees over a range of momentum values, the learn and test subsets were divided into 20 momentum bins, as already used in BaBar. These momentum bins are shown in

Table 1. A decision tree that minimized the misclassification rate was created for each bin. The type of classification tree used was Gini, which is based on a measure of node impurities [3]. Also, linear combinations of variables were allowed in splitting to enhance efficiency. For each momentum bin, three decision trees were made. One decision tree was developed using only the variables related to the Cerenkov angle and dE/dx . This tree was chosen to verify the importance of the Cerenkov angle and dE/dx as splitters over their range of effectiveness. Additionally, the accuracy of this tree was compared to results from the Neural Net. Another decision tree was developed using the same variables and Neural Net input. This tree was created to study whether decision trees used in conjunction with Neural Net results could be more effective than either method alone. A third decision tree was made using all available, potentially useful variables. The 55 additional variables primarily come from other classification methods, and all of the variables used are documented in Table 2. Since more variables were accessible in creating this decision tree, it could be expected that this would be the most accurate tree.

To compare the decision tree and Neural Net results, the sensitivity and specificity were analyzed. If there are two possible classifications of a case, type 1 and type 2, then sensitivity is defined as the number of correctly predicted cases of the target class, type 1, divided by the total number of cases of type 1. Specificity is defined as the number of cases of type 2 that are correctly classified divided by the total number of cases of type 2. For this study, specificity is the number of correctly identified pions divided by the total number of pions in the dataset. Similarly, sensitivity is defined as the number of correctly identified kaons divided by the total number of kaons in the dataset. Since the Neural Net assigns a number between 0 and 1 to each case depending on how kaon-like it is, a cutoff has to be used so that all cases with numbers greater than the cutoff will be identified as kaons. In practice, the cutoff value is chosen

depending on the application. In this study, approximately 100 different threshold values were selected and the sensitivity and specificity were calculated for each.

Decision tree testing results in a different output, but the same calculations can be performed. After testing a decision tree in CART, each terminal node of the tree contains a certain percentage of kaons. We can classify all cases in a node as kaons if the percentage of kaon cases taken from the test sample is above a certain threshold. A similar classification could be done for pions. 101 different thresholds were chosen for kaons and the specificity and sensitivity were calculated for each.

If the sensitivity is plotted against $1 - \text{specificity}$, then the resulting curve is known as the Receiver Operating Characteristic curve (ROC curve). These curves are plots of the true positive rate against the false positive rate. Necessarily, ROC curves must pass through (0,0) because if no cases are labeled positive, then both the true positive rate and the false positive rate will be 0. ROC curves must also pass through (1,1) because if every case is labeled as positive, then the true positive rate and the false positive rate will be 1. A ROC curve that is a horizontal line passing through 1 is the ideal curve and represents a perfect classification test. The area under this curve is 1. On the other hand, the 45° line represents a test where there is no relation between the test's prediction and the true classifications. The area under this curve is 0.5. Since curves that more closely resemble the ideal curve are more accurate predictors, a quality metric for performance is the area under ROC curves [4].

Using this quality metric, ROC curves were produced for each momentum bin for the decision trees and Neural Net. Examples of these ROC curves are shown in Figure 6, 7, 8, and 9. These plots show the rate of correctly classifying cases as kaons versus the rate of incorrectly classifying cases as kaons. The area under the curve is used to compare method performances.

Results

Using CART, 19 decision trees were made, corresponding to momentum bins ranging from 0.25 GeV/c to 5.0 GeV/c. No decision tree could be made using data ranging from 0 GeV/c to 0.25 GeV/c because all of the cases in the learn sample were pions, and thus no separation was possible.

Part of CART's analysis of each decision tree included information regarding the importance of each input variable used in creating the tree. For decision trees produced using only inputs related to dE/dx and the Cerenkov angle, it was noted that dE/dx was a good variable to use for momentum between 0.25 GeV/c and 0.5 GeV/c. However, dE/dx became less useful with increasing momentum up through the momentum bin, 1.125 GeV/c to 1.25 GeV/c. Then dE/dx became more useful with increasing momentum and became the most important available variable for 3.75 GeV/c through 5 GeV/c. On the other hand, the Cerenkov angle was most useful for momenta in the range, 0.75 GeV/c to 3.0 GeV/c. For decision trees produced using inputs related to dE/dx , the Cerenkov angle, and Neural Net output, the Neural Net variable was significant throughout the range of momenta studied. It was especially critical in the range 0.25 GeV/c to 0.875 GeV/c and 3.75 GeV/c to 5.0 GeV/c. Similar trends were found in the relative importance of dE/dx to the Cerenkov angle as in the trees created without the Neural Net variable. For decision trees created using all available, potentially useful variables, there were few instances of any of the previous variables mentioned being important. These trees relied primarily on variables related to other classification methods.

Using the decision tree and Neural Net results, ROC curves were produced. Examples can be seen in Figure 6, 7, 8, and 9. Out of the four curves that were plotted for each momentum

bin, it was found that the Neural Net curve was consistently the farthest from an ideal curve. For instance, in Figure 6, the Neural Net clearly has the smallest area under its curve and thus performed the worst. Furthermore, decision tree curves that used dE/dx , the Cerenkov angle, and Neural Net variables as input generally performed better than decision tree curves that did not use the Neural Net input. This was mostly noticeable in the momentum range, 0.625 GeV/c to 3.25 GeV/c, and can be seen in Figure 6 and 8. However, as shown in Figure 7, in the momentum ranges, 1.125 GeV/c to 1.25 GeV/c and 2.0 GeV/c to 2.25 GeV/c, the ROC curves using dE/dx , the Cerenkov angle, and Neural Net variables had higher sensitivity over a certain range of $1 - \text{specificity}$, while it had a lower sensitivity over other regions when compared to curves that did not use Neural Net output. Thus it is less clear which method had the best performance. Similar phenomena are evident in Figure 9. Furthermore, ROC curves of the decision trees that used all available, potentially useful variables tended to have the best performance curves in the range, 0.25 GeV/c to 1.5 GeV/c. For larger momenta, this was less clear as shown in Figure 9.

It should also be noted that the number of terminal nodes in the decision trees produced at a given momentum varied. The number of terminal nodes varied by as little as 4 for the momentum range, 4.0 GeV/c to 5.0 GeV/c, and by as many as 77 in the momentum range, 0.875 GeV/c to 1.0 GeV/c. The number of nodes for each decision tree is listed in Table 3.

Discussion and Conclusion

The results related to variable importance in the creation of decision trees using dE/dx and the Cerenkov angle were fairly consistent with the expected ability of each variable to

separate pions and kaons over the momentum range. It is expected that dE/dx would provide no separation near 1.0 GeV/c, and these results showed that dE/dx was least important in forming decision trees in the range 1.0 GeV/c to 1.5 GeV/c [1]. Also, it was expected that the Cerenkov angle provide good separation for momenta between 0.7 GeV/c and 4.0 GeV/c [5]. In this study, this variable was important in forming the decision trees for momenta in the range, 0.25 GeV/c to 3.75 GeV/c. A reason the Cerenkov angle provided good separation at lower momenta than expected is because in this range, there is a threshold below which no Cerenkov light is detected for kaons, which can be seen in Figure 2. It is the absence of this kaon data that separates the particles. These variables provided good separation even for decision trees that incorporated Neural Net results. It should further be noted that the Neural Net was especially important as an input variable for momenta greater than 3.75 GeV/c, which is consistent with the fact that both dE/dx and the Cerenkov angle provide poorer particle discrimination with increasing momentum beyond approximately 4.0 GeV/c.

Overall, the ROC curves indicated that the Neural Net classification method alone had the poorest performance. It was also found that using the decision tree with the Neural Net as one of its input variables may increase the performance of the decision tree technique, especially in the range, 0.625 GeV/c to 2.0 GeV/c. In addition, since the decision tree technique using all available, potentially useful variables tended to perform better for momenta between 0.25 GeV/c and 1.5 GeV/c, these other variables merit further investigation. Although the number of terminal nodes varied between the decision trees at a given momentum, the number of nodes did not have a consistent impact on tree performance.

This study has shown that decision trees implemented in CART are a powerful classification technique that has potential applications in particle separation. An important

advantage of this decision tree software is that it can incorporate other classification techniques' results as discriminating input variables in the design of the tree. Only input variables that are the most effective in separating the learn dataset will be used as splitters in the decision tree. Consequently, using CART in addition to other techniques can potentially increase the particle identification accuracy.

Further work can be done to supplement the results of this study. For instance, holding the momentum bin constant, a number of other variables can be varied. One example is that the data can be broken into bins based on its azimuthal angle or the polar angle and the performance of the decision tree method and Neural Net method can be compared as these bins vary. Likewise, since there are 64 potentially useful variables in the dataset related to classifying a particle as a pion or kaon, other combinations of variables can be studied.

Acknowledgements

I would like to thank the Department of Energy and Stanford Linear Accelerator Center for creating, hosting, and funding the SULI program and giving me this research opportunity. I especially want to thank my mentor, Dr. Charles C. Young, for his guidance and support.

The research described in this paper was performed at the Stanford Linear Accelerator Center during the summer of 2003.

References

- [1] K. Hagiwara et al., "Review of Particle Physics." Physical Review. D 66, 010001 (2002).

- [2] K. Swingler. *Applying Neural Networks: A Practical Guide*. San Francisco: Morgan Kaufman Publishers, Inc. (2001).
- [3] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. *Classification and Regression Trees*. Boca Raton: Chapman & Hall/CRC (1998) 2-33.
- [4] T.G. Tape. “Interpreting Diagnostic Tests.” [Online]. Available: <http://gim.unmc.edu/dxtests/Default.htm>.
- [5] L. Roos. “Recent Results with the BaBar Particle Identification System.” [Online]. Available: http://www.slac.stanford.edu/BFROOT/www/Detector/DIRC/Docs/eps99_lydia.ps (1999).

Tables

Bin #	Momentum range (GeV/c)
1	[0, 0.25]
2	[0.25, 0.5]
3	[0.5, 0.625]
4	[0.625, 0.75]
5	[0.75, 0.875]
6	[0.875, 1.0]
7	[1.0, 1.125]
8	[1.125, 1.25]
9	[1.25, 1.5]
10	[1.5, 1.75]
11	[1.75, 2.0]
12	[2.0, 2.25]
13	[2.25, 2.5]
14	[2.5, 2.75]
15	[2.75, 3.0]
16	[3.0, 3.25]
17	[3.25, 3.5]
18	[3.5, 3.75]
19	[3.75, 4.0]
20	[4.0, 5.0]

Table 1. Momentum bins used to divide the learning and testing sets of pion and kaon data.

Variable Name	Data Type	Allowed Values
charge	categorical	-1, 1
momentum	continuous	[0, +inf]
theta	continuous	[0, Pi]
phi	continuous	[-Pi, Pi]
likeKvsPi	continuous	[0, 1]
dedxdch	continuous	[0, +inf]
dedxsvt	continuous	[0, +inf]
cthe	continuous	[0, Pi]
sthe	continuous	[0, +inf]
drcxpos	continuous	[-1.8, 1.8]
svtprob	continuous	-1, [0, 1]
svtkprob	continuous	-1, [0, 1]
dchprob	continuous	-1, [0, 1]
dchkprob	continuous	-1, [0, 1]
drcprob	continuous	-1, [0, 1]
drckprob	continuous	-1, [0, 1]
ifrpiprob	continuous	-1, [0, 1]
ifrkprob	continuous	-1, [0, 1]
dedxpi	continuous	[0, +inf]
dedxk	continuous	[0, +inf]
knnval	continuous	[0, 1]
smssvtprob	continuous	-1, [0, ~5]
smssvtkprob	continuous	-1, [0, ~5]
smsdrcprob	continuous	-1, [0, ~1]
smsdrckprob	continuous	-1, [0, ~1]
ndch	ordinal	[0, ~50]
nsvt	ordinal	[0, ~16]
isvt	ordinal	[0, ~16]
fhit	ordinal	[0, 40]
lhit	ordinal	[0, 40]
nsampdch	ordinal	[0, ~35]
nsampsvt	ordinal	[0, ~10]
nphot	ordinal	[0, ~100]
nbgphot	ordinal	[0, ~100]
nphotpi	ordinal	[0, ~100]
nphotk	ordinal	[0, ~100]
drcinbar	categorical	[0, 144]
phogpi	ordinal	[0, ~100]
phogka	ordinal	[0, ~100]
isKMicroVeryLoose	categorical	0, 1
isKMicroLoose	categorical	0, 1
isKMicroTight	categorical	0, 1
isKMicroVeryTight	categorical	0, 1

isKMicroNotPion	categorical	0, 1
isKMicroNotPionGTL	categorical	0, 1
isKNNVeryLoose	categorical	0, 1
isKNNLoose	categorical	0, 1
isKNNtight	categorical	0, 1
isKNNVeryTight	categorical	0, 1
isKNNNotPion	categorical	0, 1
isKLHVeryLoose	categorical	0, 1
isKLHLoose	categorical	0, 1
isKLHtight	categorical	0, 1
isKLHVeryTight	categorical	0, 1
isKLHNotPion	categorical	0, 1
ispiLHVeryLoose	categorical	0, 1
ispiLHLoose	categorical	0, 1
ispiLHtight	categorical	0, 1
ispiLHVeryTight	categorical	0, 1
ispiRoyLoose	categorical	0, 1
ispiRoyNotKaon	categorical	0, 1
isGoodTracksVeryLoose	categorical	0, 1
isGoodTracksLoose	categorical	0, 1
isGoodTracksTight	categorical	0, 1
piK	categorical	1, 2

Table 2. The potentially useful variables included for each case in the dataset.

Momentum bin #	Number of Nodes		
	CART (all variables)	CART(dE/dx , θ_c , NN)	CART (dE/dx , θ_c)
2	46	86	82
3	61	30	91
4	57	92	107
5	74	72	58
6	108	84	31
7	66	71	52
8	75	100	91
9	179	135	103
10	169	101	105
11	88	119	80
12	123	69	57
13	89	47	66
14	51	25	30
15	77	61	36
16	54	50	42
17	22	35	24
18	29	19	29

19	13	14	26
20	17	13	15

Table 3. The Number of nodes for each decision tree made at each momentum interval.

Figures

Figure 1. Mean rate of energy change per unit distance versus momentum for various particles.

Figure 2. Cerenkov angle versus momentum for pions, kaons, and protons.

Figure 3. Distribution corresponding to the DIRC acceptance angle. The number of cases is plotted against theta. The green distribution represents kaons, and the red distribution represents pions.

Figure 4. Distribution corresponding to the D⁰ mass distribution. The number of cases is plotted against mass.

Figure 5. Distribution corresponding to the D⁰ and D*⁺ mass difference distribution. The number of cases is plotted against mass.

Figure 6. ROC curves for the momentum bin, 0.875 GeV/c to 1.0 GeV/c. The light blue line represents the Neural Net results. The yellow line represents results from the decision tree with dE/dx and the Cerenkov angle used as input variables. The pink line represents results from the decision tree with dE/dx, the Cerenkov angle, and the Neural Net data as input variables. The dark blue line represents the decision tree with all available, potentially useful variables used as input.

Figure 7. ROC curves for the momentum bin, 2.0 GeV/c to 2.25 GeV/c. The light blue line represents the Neural Net results. The yellow line represents results from the decision tree with dE/dx and the Cerenkov angle used as input variables. The pink line represents results from the decision tree with dE/dx, the Cerenkov angle, and the Neural Net data as input variables. The dark blue line represents the decision tree with all available, potentially useful variables used as input.

Figure 8. ROC curves for the momentum bin, 3.0 GeV/c to 3.25 GeV/c. The light blue line represents the Neural Net results. The yellow line represents results from the decision tree with dE/dx and the Cerenkov angle used as input variables. The pink line represents results from the decision tree with dE/dx, the Cerenkov angle, and the Neural Net data as input variables. The dark blue line represents the decision tree with all available, potentially useful variables used as input.

Figure 9. ROC curves for the momentum bin, 4.0 GeV/c to 5.0 GeV/c. The light blue line represents the Neural Net results. The yellow line represents results from the decision tree with dE/dx and the Cerenkov angle used as input variables. The pink line represents results from the decision tree with dE/dx, the Cerenkov angle, and the Neural Net data as input variables. The dark blue line represents the decision tree with all available, potentially useful variables used as input.