OPTIMIZATION OF SIGNAL SIGNIFICANCE BY BAGGING DECISION TREES

I. NARSKY
356-48 California Institute of Technology, High Energy Physics, Pasadena, CA 91125, USA
E-mail: narsky@hep.caltech.edu

An algorithm for optimization of signal significance or any other classification figure of merit (FOM) suited for analysis of HEP data is described. This algorithm trains decision trees on many bootstrap replicas of training data with each tree required to optimize the signal significance or any other chosen FOM. New data are then classified by a simple majority vote of the built trees. The performance of the algorithm has been studied using a search for the radiative leptonic decay $B \to \gamma l \nu$ at *BABAR* and shown to be superior to that of all other attempted classifiers including such powerful methods as boosted decision trees. In the $B \to \gamma e \nu$ channel, the described algorithm increases the expected signal significance from 2.4σ obtained by an original method designed for the $B \to \gamma l \nu$ analysis to 3.0σ.

1. Introduction

Various pattern classification tools have been employed in analysis of HEP data to separate signal from background. One of the problems faced by HEP analysts is the indirect nature of available classifiers. In HEP analysis, one typically wants to optimize a FOM expressed as a function of signal and background, S and B, expected in the signal region. An example of such FOM is signal significance, $S/\sqrt{S+B}$, often used by physicists to express the cleanliness of the signal in the presence of statistical fluctuations of observed signal and background. None of the available popular classifiers optimizes this FOM directly. Commercial implementations of decision trees, such as CART[1], split training data into signal- and background-dominated rectangular regions using the Gini index, $Q = 2p(1-p)$, as the optimization criterion, where p is the correctly classified fraction of events in a tree node. Neural networks[2] typically minimize a quadratic classification error, $\sum_{n=1}^{N}(y_n - f(x_n))^2$, where y_n is the true class of an event, -1 for background and 1 for signal, $f(x_n)$ is the continuous value of the neural network prediction in the range $[-1,1]$, and the sum is over N events in the training data set. Similarly, AdaBoost[3] minimizes an exponential classification error, $\sum_{n=1}^{N}\exp(-y_n f(x_n))$. These optimization criteria are not necessarily optimal for maximization of the signal significance. The usual solution is to build a neural net or an AdaBoost classifier and then find an optimal cut on the continuous output of the classifier to maximize the signal significance. Alternatively, one could construct a decision tree with many terminal nodes and then combine these nodes to maximize the signal significance.

Decision trees in StatPatternRecognition[4, 5] allow the user to optimize any FOM supplied as an implementation of an abstract C++ interface included in the package. A default implementation of the decision tree includes both standard figures of merit used for conventional decision trees such as the Gini index and HEP-specific figures of merit such as the signal significance or the signal purity, $S/(S+B)$.

A decision tree, even if it directly optimizes the desired FOM, is rarely powerful enough to achieve a good separation between signal and background. The mediocre predictive power of a single decision tree can be greatly enhanced by one of the two popular methods for combining classifiers — boosting[3] and bagging[6]; the latter approach can be used in conjunction with the random forest technology[7]. This note compares predictive power of several classifiers using a search for the radiative leptonic decay $B \to \gamma l \nu$ at BaBar. It is shown that the greatest signal significance is obtained by bagging an ensemble of decision trees, with each member of the ensemble optimizing the signal significance. This study is described in more detail in two notes[4, 5] posted at the physics archive.

2. Decision Trees in StatPatternRecognition

A decision tree recursively splits training data into rectangular regions (nodes). For each node, the tree examines all possible binary splits in each dimension and selects the one with the highest FOM. This procedure is repeated until a stopping criterion, specified as the minimal number of events per tree node, is satisfied. The tree continues making new nodes until it is composed of leaves only — nodes that cannot be split without a decrease in the FOM and nodes that cannot be split because they have too few events.

As mentioned above, a conventional decision tree often uses the Gini index, $Q(p,q) = -2pq$, for split optimization, where p and $q = 1 - p$ are fractions of correctly classified and misclassified events in a given node. If a parent node with the total event weight W is split into two daughter nodes with weights W_1 and $W_2 = W - W_1$, the best decision split is chosen to maximize $Q_{\text{split}} = (W_1 Q_1 + W_2 Q_2)/W$, where Q_1 and Q_2 are figures of merit computed for the two daughter nodes. Note that a conventional decision tree treats the two categories, signal and background, symmetrically. In HEP analysis, one usually wishes to optimize an asymmetric FOM. StatPatternRecognition offers a modified splitting algorithm for this purpose. The best decision split is now chosen to maximize $Q_{\text{split}} = \max(Q_1, Q_2)$, where Q_1 and Q_2 are the asymmetric figures of merit for the daughter nodes. In case of the signal significance, the FOM is given by $Q(s,b) = s/\sqrt{s+b}$, where s and b are signal and background weights in a given node. After the tree is grown, the terminal nodes are merged to optimize the overall asymmetric FOM. The merging algorithm sorts all terminal nodes by signal purity in descending order and computes the overall FOM for the n first nodes in the sorted list with n taking consecutive values from 1 to the full length of the list. The optimal combination of the terminal nodes is given by the highest FOM computed in this manner.

This algorithm for optimization of an asymmetric FOM is nothing but an empirical solution. It is not guaranteed that this algorithm will produce a higher asymmetric FOM than the one obtained by a conventional decision tree using the Gini index or any other symmetric expression as a split criterion. It has been shown experimentally that this algorithm tends to produce higher values of the signal significance when applied to physics data sets. This note is an example of such an application.

3. Bagging Decision Trees

The predictive power of a single classifier can be enhanced by boosting[3] or bagging[6]. Both these methods work by training many classifiers, e.g., decision trees, on variants of the original training data set. A boosting algorithm enhances weights of misclassified events and reduces weights of correctly classified events and trains a new classifier on the reweighted sample. In contrast, bagging algorithms do not reweight events. Instead, they train new classifiers on bootstrap replicas of the training set. After training is completed, events are classified by the majority vote of the trained classifiers. For successful application of the bagging algorithm, the underlying classifier must be sensitive to small changes in the training data. Otherwise all trained classifiers will be similar, and the performance of the single classifier will not be improved. This condition is satisfied by a decision tree with fine terminal nodes. Because of the small node size each decision tree is significantly overtrained; if the tree were used just by itself, its predictive power on a test data set would be quite poor. However, because the final decision is made by the majority vote of all the trees, the algorithm delivers a high predictive power.

Random forest[7], typically used in conjunction with bagging, is a technique that randomly selects a subset of input variables for each decision split. This approach can make individual trees more independent of each other and increase the overall predictive power.

Boosting and bagging algorithms offer competitive predictive power. It is really hard, if possible, to predict outright which algorithm will perform better in any classification problem. For optimization of the signal significance, however, bagging is the choice favored by intuition. Reweighting events has an unclear impact on the effectiveness of the optimization routine with respect to the chosen asymmetric FOM. While it may be possible to design a reweighting algorithm efficient for optimization of a specific FOM, at present such reweighting algorithms are not known. Bagging, on the other hand, offers an obvious solution. If the base classifier directly optimizes the chosen FOM, bagging is equivalent to optimization of this FOM integrated over bootstrap replicas.

4. Separation of Signal and Background in a Search for the Radiative Leptonic Decay $B \to \gamma l \nu$ at BaBar

A search for the radiative leptonic decay $B \to \gamma l \nu$ is currently in progress at BaBar; results of this analysis will be made available to the public in the near future. The analysis focuses on measuring the B meson decay constant, f_B, which has not been previously measured.

Several samples of simulated Monte Carlo (MC) events are used to study signal and background signatures in this analysis: $B \to \gamma l \nu$ signal samples with about 1.2M events in each channel, large samples of generic B^+B^-, $B^0\bar{B}^0$, $c\bar{c}$, uds and $\tau^+\tau^-$ MC events, as well as several exclusive semileptonic modes generated separately with a typical sample size of several hundred thousand events.

Various preliminary requirements have been imposed to enhance the signal purity and at the same time reduce the MC samples to a manageable size. After these preliminary requirements have been imposed, eleven variables are included in the final optimization procedure. Distributions of these variables and more details on applied selection requirements can be found elsewhere[4].

The signal and combined background MC samples are used by various optimization algorithms to maximize the signal significance expected in 210 fb^{-1} of data. The training samples used for this optimization consist of roughly half a million signal and background MC events in both electron and muon channels, appropriately weighted according to the integrated luminosity observed in the data. The training:validation:test ratio for the sample sizes is 2:1:1. Signal MC samples are weighted assuming a branching fraction of 3×10^{-6} for each channel.

The authors of this analysis deploy an original cut optimization routine[4] for separation of signal and background. This procedure divides the available range for each variable into intervals of preselected length and finds an optimal set among all possible combinations of orthogonal cuts. Besides the original method designed by the analysts, several classifiers have been used:

- Decision tree optimizing the signal significance $S/\sqrt{S+B}$.
- Bump hunter[8] optimizing the signal significance.
- 700 boosted binary splits.
- 50 boosted decision trees with minimal node size 100 events.
- Combiner of subclassifiers trained on individual background components using boosted binary splits.
- 100 bagged decision trees with each tree optimizing the signal significance; the minimal node size has been set to 100 events.

Parameters of all classifiers have been optimized by comparing values of the statistical significance obtained for the validation samples.

Results are shown in Table 1. The output of the described bagging algorithm for the $B \to \gamma e \nu$ test data is shown in Fig. 1. The bagging algorithm provides the best value of the signal significance. It gives a 24% improvement over the original method developed by the analysts, and a 14% improvement over boosted decision trees; both numbers are quoted for the $B \to \gamma e \nu$ channel.

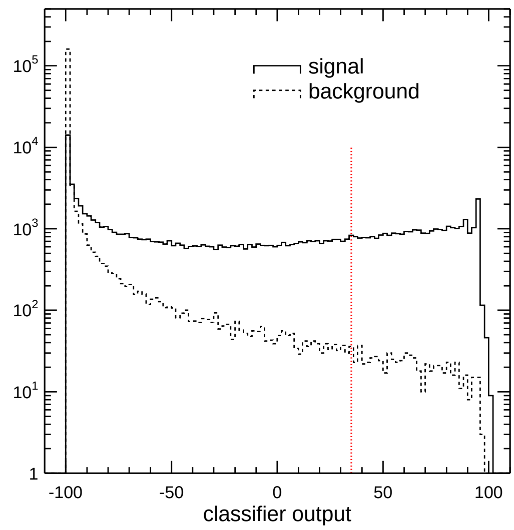

Fig. 1. Output of the bagging algorithm with 100 trained decision trees for the $B \to \gamma e \nu$ test sample. The cut maximizing the signal significance, obtained using the validation sample, is shown with a vertical line.

The bagging algorithm with decision trees optimizing the Gini index showed an 8% improvement in the $B \to \gamma e \nu$ signal significance compared to the boosted decision trees. But the signal signifi-

Table 1. Signal significances, $\mathcal{S}_{\text{train}}$, $\mathcal{S}_{\text{valid}}$, and $\mathcal{S}_{\text{test}}$, for the $B \to \gamma l \nu$ training, validation, and test samples obtained with various classification methods. The signal significance computed for the test sample should be used to judge the predictive power of the included classifiers. W_1 and W_0 represent the signal and background, respectively, expected in the signal region after the classification criteria have been applied; these two numbers have been estimated using the test samples. All numbers have been normalized to the integrated luminosity of 210 fb^{-1}. The best value of the expected signal significance is shown in boldface.

Method	$B \to \gamma e \nu$					$B \to \gamma \mu \nu$				
	$\mathcal{S}_{\text{train}}$	$\mathcal{S}_{\text{valid}}$	$\mathcal{S}_{\text{test}}$	W_1	W_0	$\mathcal{S}_{\text{train}}$	$\mathcal{S}_{\text{valid}}$	$\mathcal{S}_{\text{test}}$	W_1	W_0
Original method	2.66	-	2.42	37.5	202.2	1.75	-	1.62	25.8	227.4
Decision tree	3.28	2.72	2.16	20.3	68.1	1.74	1.63	1.54	29.0	325.9
Bump hunter with one bump	2.72	2.54	2.31	47.5	376.6	1.76	1.54	1.54	31.7	393.8
Boosted binary splits	2.53	2.65	2.25	76.4	1077.3	1.66	1.71	1.44	45.2	935.6
Boosted decision trees	13.63	2.99	2.62	58.0	432.8	11.87	1.97	1.75	41.6	523.0
Combiner of background subclassifiers	3.03	2.88	2.49	83.2	1037.2	1.84	1.90	1.66	55.2	1057.1
Bagged decision trees	9.20	3.25	**2.99**	69.1	465.8	8.09	2.07	**1.98**	49.4	571.1

cance obtained with this method was 9% worse than that obtained by the bagging algorithm with decision trees optimizing the signal significance. The 14% improvement of the proposed bagging algorithm over the boosted decision trees therefore originated from two sources: 1) using bagging instead of boosting, and 2) using the signal significance instead of the Gini index as a FOM for the decision tree optimization.

In an attempt to improve the signal significance even further, the random forest approach has been attempted with the number of randomly sampled (with replacement) input variables taking values 1, 6, and 11. No significant improvement over the bagging algorithm has been found.

This note describes a somewhat unusual application of boosted and bagged decision trees to data analysis with the ultimate goal of classification defined as maximization of the signal significance. The classifier performance in this case is driven by a small fraction of the data set included in the signal region. In a typical application of boosted decision trees, one minimizes the exponential loss averaged over the whole data set. The optimal node size for boosted decision trees is typically much larger than the optimal node size for bagged decision trees. In this analysis, the optimal node sizes for both boosted and bagged decision trees are comparable.

5. Summary

A bagging algorithm suitable for optimization of an asymmetric FOM for HEP analyses has been described. This algorithm has been shown to give a significant improvement of the signal significance in the search for the radiative leptonic decay $B \to \gamma l \nu$ at *BaBar*.

Acknowledgments

Thanks to Gregory Dubois-Felsmann, Byron Roe and Frank Porter for useful discussions and comments on this work. Thanks to Ed Chen for data and documentation on the $B \to \gamma l \nu$ analysis. Thanks to Harrison Prosper for presenting this work at Phystat 2005. This work is partially supported by Department of Energy under Grant DE-FG03-92-ER40701.

References

1. L. Breiman et al., *Classification and Regression Trees*, Waldsworth International, 1984.
2. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 1999.
3. Y. Freund and R.E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, J. of Computer and System Sciences **55**, 119-139 (1997).
4. I. Narsky, *StatPatternRecognition: A C++ Package for Statistical Analysis of High Energy Physics Data*, physics/0507143, 2005.
5. I. Narsky, *Optimization of Signal Significance by Bagging Decision Trees*, physics/0507157, 2005.
6. L. Breiman, *Bagging Predictors*, Machine Learning **26**, 123-140 (1996).
7. L. Breiman, *Random Forests*, Machine Learning **45**, 5-32 (2001).
8. J. Friedman and N. Fisher, *Bump hunting in high dimensional data*, Statistics and Computing **9**, 123-143 (1999).