Z Splitting Criterion for Growing Trees and Boosting

Harris Drucker, Monmouth University,
West Long Branch, NJ 07764

Abstract

A splitting criterion that arrives out of the context of a new boosting algorithm is used to construct classification trees. Trees constructed using this Z function are compared to those using the entropy function of C4.5 and are found to give much lower error rates. The Z function is also used to construct boosting machines which when compared to other implementations give lower error rates.

Keywords: classification trees, boosting, C4.5

Introduction

One aspect of this paper is a discussion of a new splitting criterion, called the Z function which can be used to build classification trees as opposed to the splitting criteria of C4.5 [2] and CART [18] and others discussed by Mingers [17]. The second objective of this paper is to describe a new procedure to combine trees in a boosting ensemble of trees that has much better performance than a single classification tree. Boosting as a learning algorithm was initiated by Schapire [19] and many variations now exist [1, 3-12, 14, 15, 21]. In the following sections we discuss a new boosting algorithm that applies no matter what the form of the "weak learner" but here we concentrate on binary classification trees. A "weak learner" is a learning algorithm that is guaranteed to have an error rate of less than 1/2.

Boosting builds an ensemble of weak learners, each weak learner in the ensemble concentrating on those members of the training set that prior members of the ensemble misclassified. Figure 1 (a modification from [20]) shows the training algorithm. Weak learner t (in our case a classification tree) either classifies pattern i correctly or not. $D_{t+1}(i)$ then becomes larger for incorrectly classified patterns and as we build our ensemble of T weak learners, more emphasis is placed on correctly classifying those patterns that have larger $D_{t+1}(i)$. Usually T is not chosen a priori but rounds of boosting continued until the training error rate is zero (and sometimes beyond). The denominator of the equation in step three is:

$$Z_t = \sum_{i=1}^{m} D_t(i)\exp(-y_i h_t(i)) \qquad (1)$$

and is crucial to defining a new splitting criterion for trees because we can show [20] that minimizing Z minimizes the training error. Thus we shall attempt to minimize Z for each tree. This function has been investigated elsewhere [13] but did not show the improvement we show here. We attribute this to the different pruning procedures described later.

Growing Trees Using the Z Function

Trees are constructed using a training set and pruned using a pruning set. Trees consist of a series of nodes each with connections to two nodes below unless the node is a terminal node (termed a leaf). The root node examines one of the input features, and determines whether the value of that feature is greater than some critical value x_c. If greater than x_c the pattern follows the right branch of the tree, else the left branch. This process is repeated for every node traversed until a terminal node is reached and the classification of the input is via the terminal node

---

Given: Training m training examples with classifications: $y \in Y = \{-1,1\}$. Initialize $D_i = 1/m$

For $t = 1,\ldots,T$

1. Train weak learner using distribution D_t
2. Get weak hypothesis h_t
3. Update:

$$D_{t+1}(i) = \frac{D_t(i)\exp(-y_i h_t(i))}{Z_t}$$

where Z_t is a normalization factor (chosen such that D_t is a distribution

Output the final hypothesis:

$$H(x) = sign\left(\sum_{i=1}^{T} h_t(x_i)\right)$$

Figure 1: The AdaBoost Training Algorithm [20]

reached. A node is specified by one of the features, a critical value of that feature, and the hypothesis generated when a sample reaches that node. Let us examine a particular feature for all the examples in the root node and let W_+^L be the sum of all D(i) that have a value of the feature $x \le x_c$ and have class +1 (L stands for the left node immediately below and to the left of the root node). Let c_L be the hypothesis of the node immediately below and to the left of the root node. All the examples in the root node with a feature less than this critical value x_c would then end up in the L node. Similarly we define W_-^L as the sum of the D(i) in the root node whose feature value is less than the critical value and class is -1. Then we can rewrite equation (1) as:

$$Z = \{W_+^L e^{-c_L} + W_-^L e^{c_L}\} + \{W_+^R e^{-c_R} + W_-^R e^{c_R}\} \quad (2)$$

with the obvious extension to the R node. The first term (between braces) can be minimized by choosing $c_L = \frac{1}{2}\ln\left(\frac{W_+^L}{W_-^L}\right)$ and a similar expression for the second term. These would be the hypotheses for the left and right nodes below the root node if this particular feature was used and this critical value of this particular feature was used. These hypotheses may be large negative or positive if the majority of the examples that would arrive in a node have a larger summation of the D(i) in one class versus the other class. On the other hand, if the sum of the D(i) from each of the two classes are approximately equal, then the hypotheses are close to zero, indicating low confidence in the hypothesis. This is an example of a soft decision process and is opposite in philosophy to C4.5 and CART which tend to make hard decisions (that it, the hypothesis is either +1 or -1).

Thus, recursively, for each node we examine all the features and all the instances of that feature to find the critical value x_c and the feature that minimizes equation (2). We continue this until there are five or less examples in a node. The rationale for the value "five" is that we prune the tree back later and nodes that have few examples will tend to be eliminated anyhow. It may be the case that all the examples in a particular node belong to one class in which case the hypothesis is plus or negative infinity. In those cases, we add a smoothing constant so that $c_L = \frac{1}{2}\ln\left(\frac{W_+^L + \varepsilon}{W_-^L + \varepsilon}\right)$ where $\varepsilon = 1/2m$ and m is the number of training examples.

Building trees using the approach described above builds trees with very small training error. However, we are interested in building trees with small testing error. Therefore, we prune [16] the tree to give better generalization by examining all the terminal nodes and working our way up the tree towards the root node, removing nodes as we go.

In the expression below, the P's correspond to the analog of the W's above except that P refers to a separate pruning set of size 20% of the training set. Let A, L, and R indicate the node "Above", the "Left", and "Right" nodes respectively. We remove the L and R nodes and make the A node a terminal node if:

$$P_+^A e^{-c_A} + P_-^A e^{c_A} \le P_+^L e^{-c_L} + P_-^L e^{c_L} + P_+^R e^{-c_R} + P_-^R e^{c_R} \quad (3)$$

The basic rationale for this is that using (2), one is guaranteed that the sum of the Z's for the two nodes immediately below a parent node is less than the Z of the parent node. One would hope, for good generalization, that any other set of patterns would have similar characteristics. Using (3), one guarantees this is true for the pruning patterns. Note also that the hypotheses c_j (j is either A, L, or R) comes from the training set, not the pruning set.

Experiments

The first set of experiments consists of classifying patterns from a NIST (National Institute of Standards and Technology) database of 120,000 digits where digits 0 to 4 are assigned as class +1 and the other digits as class -1. We wanted a method to obtain multiple data sets of increasing difficulty. To implement this, we subsampled the original data to be of size 10 by 10 (100 features). We then train a single layer neural network so that it acts as a filter producing difficult and easy classification problems. Let us call the output of this network h_{single}. This network is trained using 20,000 examples and has a test error rate of 18%. Examples classified correctly by this neural network are considered easy examples and those classified incorrectly are hard examples and by mixing the numbers of each type we can vary the fraction of difficulty f.

Specifically, we iterate the below until we have enough examples (training, pruning, and testing):

- Input pattern i from the NIST database with classification h_{single} and correct class y_i.

- If $h_{single} \neq y_i$ then accept sample i with probability f, else accept sample i with probability $(1-f)$.

Table I shows the ensemble and single tree test error rates for training sets of size 10,000 and 1,000 using both C4.5 and Z as the splitting function. In each of these cases the pruning set is 20% of the size of the training set and the test set is of size 20,000. The results are averaged over ten runs. In each run, in comparing Z to C4.5, we use the same training, pruning, and test sets and in all cases, using Z was statistically significantly better than C4.5.

Table II shows some results from the University of California at Irvine (UCI) repository. In addition to our results, we also list the best (F-S) results from Freund and Schapire [10]. In those case we use ten-fold cross validation repeated ten times for a total of 100 runs limited to 100 trees per boosting ensemble where 90% of the total samples is divided into a pruning set and training set. Once again the Z function is better than C4.5. Comparing the best Z results to the last column, we see a virtual tie in one case and large improvements in the two other cases.

Conclusions

A splitting criterion based on the Z function gives significantly better performance when constructing single trees and boosting ensembles as compared to using the splitting criterion of C4.5. In both of these cases, pruning is essential in obtaining good generalization.

Acknowledgements

The help of Robert Schapire and Yoram Singer is gratefully acknowledged. The use of the UCI database is also acknowledged.

1,000 training examples					10,000 training examples			
degree of difficulty f	Single tree C4.5	Single tree Z	Boosted Ensemble C4.5	Boosted Ensemble Z	Single tree C4.5	Single tree Z	Boosted Ensemble C4.5	Boosted Ensemble Z
.1	16.9	12.7	6.56	4.59	10.3	7.77	2.16	1.97
.3	20.2	14.8	7.87	6.30	12.4	10.3	2.92	2.17
.5	23.4	17.1	11.2	7.73	13.7	10.7	3.77	3.12
.7	31.1	20.9	17.4	9.52	16.7	12.6	5.15	3.76
.9	37.7	24.8	22.6	11.4	20.8	14.7	5.95	4.42

Table 1. Error rate in per cent for digit database

Database	Total Patterns	Single Tree C4.5	Single Tree Z	Boosted Ensemble C4.5	Boosted Ensemble Z	F-S Best Results
breast cancer-W	699	6.20	4.51	2.47	2.33	3.2
pima Indians	768	34.3	30.0	27.6	24.5	24.4
ionosphere	351	10.2	8.22	4.64	4.56	5.8

Table 2. Error rate in per cent for UCI database

References

[1] Leo Breiman, "Prediction games and arcing classifiers", Technical Report 504, Statistics Department, University of California at Berkeley, 1997.

[2] Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles Stone, *Classification and Regression Trees*, Wadsworth International Group, 1984.

[3] T. Dietterich, M. Kearns, and Y. Mansour, "Applying the weak learning framework to understand and improve C4.5", *Machine Learning: Proceedings of the International Conference on Machine Learning*, Morgan Kaufmann, 1996.

[4] Harris Drucker, "Improving regressors using boosting techniques", *Proceedings International Conference on Machine Learning*, pages 107-115, Morgan Kaufmann, 1997.

[5] Harris Drucker, Corinna Cortes, L.D. Jackel, Yann LeCun, and Vladimir Vapnik, "Boosting and other ensemble methods", *Advances in Neural Information Processing Systems 8*, eds: David S. Touretsky, Michael C. Mozer, and Michael E., Hasselmo, pages 479-485, Morgan Kaufmann, 1996.

[6] Harris Drucker, Robert Schapire, and Patrice Simard, "Boosting performance in neural networks, *International Journal*

of Pattern Recognition and Artificial Intelligence, 7(4):705-719, 1993.

[7] Harris Drucker, Robert Schapire and Patrice Simard, "Improving performance in neural networks using a boosting algorithm", *Advances in Neural Information Processing Systems 5*, eds: Stephen Jose Hanson, Jack D. Cowan, and C. Lee Giles, pages 42-49, Morgan Kaufmann, 1993.

[8] Yoav Freund, "Boosting a weak learning algorithm by majority", *Proceedings of the Third Workshop on Computational Learning Theory*, pages 202-216, Morgan Kaufmann.

[9] Yoav Freund and Robert E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *Computational Learning Theory: Second European Conference EuroCOLT '95*, pages 23-37, Springer Verlag, 1995.

[10] Yoav Freund and Robert E. Schapire, "Experiments with a new boosting algorithm", *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148-156, 1996.

[11] Yoav Freund and Robert Schapire, "Game Theory, on-line prediction and boosting", *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 325-332, 1996.

[12] Yoav Freund and Robert Schapire, " A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of Computer and System Science*, 55(1):119-139.

[13] Michael Kearns and Yishay Mansour, "On the boosting ability of top-down decision trees learning algorithms", *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Morgan Kaufmann*, 1996.

[14] Y. LeCun, L.D. Jackel, L. Bottou, A. Brunot. C. Cortes, J.S. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, and V. N. Vapnik, "Comparison of learning algorithms for handwritten digit recognition", eds: F. Fogelman and P. Gallinari, *International Conference on Artificial Neural Network*, pages 53-60, Paris, 1995, EC2 & Cie.

[15] Y. LeCun, L.D. Jackel, L. Bottou, A. Brunot. C. Cortes, J.S. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, and V. N. Vapnik, "A comparison on handwritten digit recognition", eds: L J. H. Oh, C. Kwon, and S. Cho, *Neural Networks: The Statistical Mechanics Perspective*, World Scientific, 1995, pages 261-276.

[16] J. Mingers, "An empirical comparison of pruning methods for decision trees", *Machine Learning*, "4:227-243, 1989.

[17] J. Mingers, "An empirical comparison of selection methods for decision tree induction", *Machine Learning*, 4:319-342.

[18] J. Ross Quinlin, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1988.

[19] Robert E. Schapire, "The strength of weak learnability", 30th *Annual Symposium on Foundations of Computer Science*, pages 28-33, October, 1989.

[20] Robert E. Schapire and Yoram Singer, "Improved boosting algorithms using confidence-rated predictions". *Proceedings International Conference on Machine Learning*, 1998.

[21] Holger Schwenk and Yoshua Bengio, "Adaptive boosting of neural networks for character recognition". Advances in Neural Information Processing 10, 1997,

INTERNATIONAL NEURAL NETWORK SOCIETY

423-3135

March 31, 1999

Harris Drucker
Monmouth University
West Long Branch NJ 07764
USA

Dear Dr. Drucker:

On behalf of the International Neural Network Society and the Neural Networks Council, IEEE, I would like to congratulate you on the acceptance of your paper and thank you for agreeing to participate in IJCNN'99, July 10-16, 1999 at the Renaissance Hotel in Washington, D.C. Your paper has been accepted for **ORAL** presentation:

Paper # and Title: 184 - Z Splitting Criterion for Growing Trees and Boosting
Session: 5.4

Further information concerning the meeting schedule and registration will be posted on the IJCNN'99 web site: http://www.cas.american.edu/~medsker/ijcnn99/ijcnn99.html or reachable via the INNS site: http://www.inns.org/.

Please electronically submit your completed four page paper according to the format information on the IJCNN'99 website to the publications chair, Jonathan Boswell: ijcnn99@ost.cdrh.fda.gov, by May 10, 1999. Submission of your paper implies acceptance of transfer of copyright ownership in the above titled paper to IEEE, subject to the extent of applicable national law and to the reasonable use of authors or their employers. Please complete the enclosed copyright form and return by fax (609-423-3420) to the IJCNN office no later than May 10, 1999.

The presenter's registration must be submitted by May 10, 1999. Each presenter must register for the meeting and is required to pay the registration fee as well as hotel and travel expenses. Papers will not be listed in the program nor published in the proceedings unless the registration fee is paid.

I look forward to your participation in IJCNN'99. Should you have any questions regarding your participation, please do not hesitate to contact IJCNN at 609-423-7222 ext. 350, by e-mail at innsmtg@talley.com or by fax 609-423-3420.

Best regards,

David G. Brown
dp

David G. Brown
General Chair

19 Mantua Road, Mount Royal, New Jersey, 08061, USA
Phone: 609-423-0162 Fax: 609-423-3420
E-Mail: innshq@talley.com
WWW: inns.org