Boosted ARTMAP

Stephen J. Verzi [†], Gregory L. Heileman[‡], Michael Georgiopoulos[*], Michael J. Healy[**]

[†] Department of Computer Science
verzi@cs.unm.edu
[‡] Department of Electrical and Computer Engineering
University of New Mexico, Albuquerque, NM 87131
heileman@eece.unm.edu

[*] Department of Electrical and Computer Engineering
University of Central Florida, Orlando, FL 32816
mng@ece.engr.ucf.edu

[**] The Boeing Company
P.O. Box 3707 MS 7L-66, Seattle, WA 98124
mjhealy@boeing.com

Abstract

We present a modification to the Fuzzy ARTMAP neural network architecture for conducting boosted learning in a probabilistic setting. We call this new architecture boosted ARTMAP (BARTMAP). Performance comparison with Fuzzy ARTMAP, PROBART and ART-EMAP on some simple two-class problems is discussed. Experimental results indicate that BARTMAP gives better generalization results on some problems involving classification overlap. In addition BARTMAP requires fewer resources, i.e., network nodes, to achieve performance levels comparable to those in Fuzzy ARTMAP.

1. Introduction

An important performance measure of a machine learning algorithm is its generalization capability. Generalization is characterized by the number of unseen examples correctly predicted by a learning algorithm given sample training data from which to learn. One way of increasing a learning algorithm's generalization ability is to reduce its error on training data while providing it training data highly representative of the unknown target function. Boosting is a technique designed to improve a learning algorithm's performance and generalization across the distribution of examples, especially in areas where the particular algorithm is having difficulty [1].

Fuzzy ARTMAP is a neural network architecture for conducting supervised learning in a multidimensional setting [2, 3]. When Fuzzy ARTMAP is used on a learning problem, it is trained to the point that it correctly classifies all training data. This feature causes ARTMAP to "overfit" some data sets, especially those with overlap. To avoid the problem of "overfitting", we must allow for error in the training process. One solution for allowing error during the training is to use a statistical approach.

A number of other architectures have been proposed that introduce a stochastic element to Fuzzy ArtMap training. For example, PROBART is a specialization of Fuzzy ARTMAP with probabilistic learning capabilities useful in regression learning [4], and ART-EMAP is an extension of ARTMAP that includes evidence accumulation [5]. Modified FAM is another extension of the Fuzzy ARTMAP category formation and selection processes aimed at minimizing misclassification rates [6].

In this paper, we will present a modification to Fuzzy ARTMAP which also employs statistical techniques. The design of our architecture, called BARTMAP, was motivated by the theory of boosting. Kearns and Mansour demonstrate the boosting capabilities of CART and C4.5 [7]. In their paper, computational complexity bounds as well as generalization capabilities for decision tree algorithms are detailed. The promise of theoretical bounds on computational complexity as well as generalization capabilities for ART-based architectures motivates the research described in this paper.

In section 2, we describe Fuzzy ART and Fuzzy ARTMAP. In section 3, we describe BARTMAP. Empirical results are presented in section 4, and conclusions are discussed in section 5.

2. Fuzzy ART and Fuzzy ARTMAP

The Fuzzy ART neural network architecture was designed to cluster data into categories. Fuzzy ART is structured into three layers of interacting neural nodes, labeled

0-7803-4859-1/98 $10.00©1998 IEEE

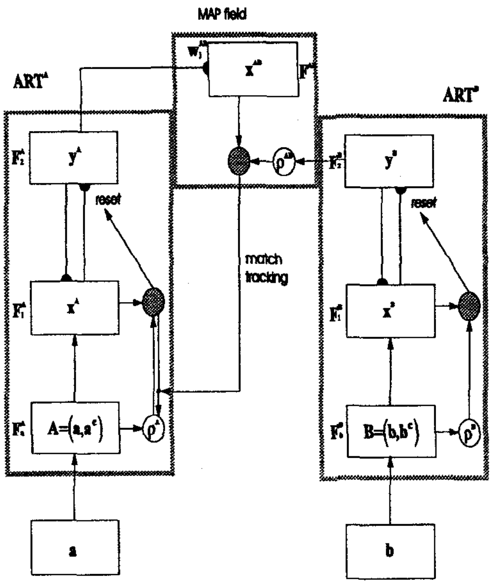

Figure 1: The Fuzzy ARTMAP Architecture.

F_0, F_1 and F_2, where the output of F_0 is connected to F_1, and F_1 and F_2 are mutually connected. At F_0, input is received from the environment in the form of an M-length vector. At this point, the input is complement coded producing a $2M$-length vector, I, which is passed on to F_1.

The F_1 and F_2 layers interact to choose an F_2 template that best matches the complement coded input vector according to:

$$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|}. \tag{1}$$

This choice is confirmed if the vigilance criterion is not violated, i.e.,

$$\frac{|I \wedge w_j|}{|I|} \geq \rho. \tag{2}$$

The parameter α is called the choice parameter and is usually a positive but small quantity, and ρ is a user input between 0 and 1 where a value closer to 1 indicates desired tighter coupling within clustered patterns and a value closer to 0 allows less coupling within clustered patterns.

The process of complement coding a pattern vector, a, produces a new vector $A = (a, a^c)$, where a^c is the complement of a. There are two stages in ART cluster formation. A winner-take-all strategy is employed in choosing the best matching cluster template in the F_2 layer given a complement coded input vector according to (1). Next, a vigilance check is performed to ensure that learning the input pattern in the chosen cluster will not degrade the template below the vigilance as in (2). Initially all template weights are set to 1, and learning proceeds as follows

$$w_j^{(new)} = \beta\left(I \wedge w_j^{(old)}\right) + (1-\beta)w_j^{(old)},$$

where β is the learning parameter. In this paper we will set $\beta = 1$ which is a special case called fast learning.

An important feature of ART is that the F_2 layer is allowed to grow as needed for a particular problem. A pool of templates is maintained, where a committed template has, at some point, learned at least one input pattern. A committed template is always preferred in the choice stage of clustering, but if a particular input pattern fails the vigilance test for all committed templates, then a new uncommitted template is chosen to learn this pattern. Once a template learns a pattern it becomes committed.

The Fuzzy ARTMAP architecture consists of two Fuzzy ART modules connected through a MAP field. The ART^A module is given pattern data and the ART^B module is given label data for a given supervised learning task. The MAP field links pattern clusters with associated label clusters. Supervised learning is performed in Fuzzy ARTMAP by ensuring that each ART^A template associates with only one ART^B template. Thus, a many-to-one mapping from patterns to labels will be formed.

During supervised learning with Fuzzy ARTMAP, each (pattern, label) pair, presented to the network, represents a correctly labeled pattern. If a pattern presented to the ART^A module chooses a template associated with a different label than presented to the ART^B module, ARTMAP will perform a lateral reset. The lateral reset forces the ART^A module to choose a different template, by raising the vigilance. The vigilance value will remain high during the presentation of this data pair. Initially uncommitted templates in both ART^A and ART^B have no association in the MAP field.

In this paper, we are interested in concept learning, specifically 0-1 concept learning. Here, a 0 label indicates that the pattern is a negative example of the concept, and a 1 label means that the pattern is a positive example of the concept. Thus, for the purposes of this paper, the ART^B module will contain only two templates, one for 0 and one for 1.

3. Boosted ARTMAP (BARTMAP)

Similar to Fuzzy ARTMAP, BARTMAP is composed of two BART modules, defined below, connected by a MAP field. The BART module is an extension of the Fuzzy ART module, and BARTMAP is an extension of PROBART which is a modification of Fuzzy ARTMAP.

The BART module is exactly the same as an Fuzzy ART module except that instead of having a single vigilance parameter, each F2 node has its own vigilance. Since each cluster template has its own vigilance parameter, the

2 2

ρ_i, instead of a single ρ for the entire module

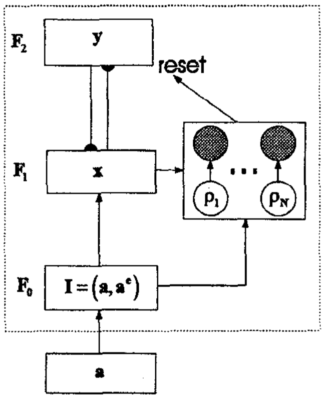

Figure 2: The BART Module.

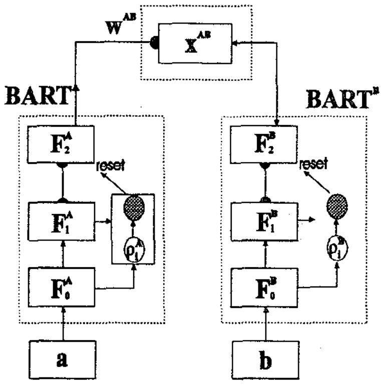

Figure 3: The BARTMAP Architecture.

as in ARTMAP.

The change to Fuzzy ART was motivated by a desire to allow the categories formed during learning to define their own sizes. In Fuzzy ARTMAP, the vigilance parameter is the major factor controlling the sizes of clusters formed.

The clustering metric for Fuzzy ART actually includes two separate competing functions. The choice portion of the metric favors adding a new instance to a more tightly coupled category, and within the same degree of coupling, it favors those categories with highest magnitude. The vigilance portion of the metric tests to make sure that inclusion of the new instance into the chosen category does not violate the spatial extent constraints. In other words, the new instance is not allowed to degrade or lower the magnitude of the category template below the vigilance.

The modification to Fuzzy ART proposed in BARTMAP allows each cluster to develop according to its own vigilance instead of a common vigilance as in Fuzzy ARTMAP. The idea here is to allow each category to "cover" its portion of the data according to the underlying distribution.

BARTMAP is designed to conduct supervised concept learning in a probabilistic setting. The MAP field in Fuzzy ARTMAP is used to ensure that each template on the A side is associated with only one template on the B side, a many-to-one mapping. In BARTMAP, the MAP field is used to track the frequency of associations between $BART^A$ and $BART^B$ templates, as is done in PROBART. In PROBART, however, there is no way of controlling which ART^A templates are linked with particular ART^B templates. During supervised learning in BARTMAP, the vigilance parameter for a specific $BART^A$ template is increased in order to decrease the number of different associations between this template and $BART^B$ templates.

The map field weight vector is modified during learning to directly count the number of associations between $BART^A$ and $BART^B$ templates. Initially all map field weights are set to 0, and during learning

$$w_{JK}^{AB} = w_{JK}^{AB} + 1 \tag{3}$$

where J indexes the template from the $BART^A$ module chosen by the pattern, and K indexes the template from the $BART^B$ module chosen by the label.

In BARTMAP, each $BART^A$ template can be associated with many, even all, $BART^B$ templates. The label predicted for a specific $BART^A$ template is the $BART^B$ template with the largest association. Given a randomly generated example, (x, y), the error of BARTMAP, ε_T will be defined as the product of the probability that x chooses $BART^A$ template J and the probability that y is not the predicted label for $BART^A$ template J, summed over all $BART^A$ templates. The error is estimated using the frequency information obtained in w^{AB} and the training sample size, S. Thus, given a trained BARTMAP architecture, the error for a random example is calculated as

$$Pr\left\{x \text{ chooses } \nu_J^A\right\} = \frac{|w_J^{AB}|}{S}, \tag{4}$$

$$Pr\left\{y \text{ not predicted by } \nu_J^A\right\} = 1 - \frac{\max_k \left\{w_{Jk}^{AB}\right\}}{|w_J^{AB}|}, \tag{5}$$

$$\varepsilon_T = \frac{\sum_{j=1}^{N^A} \left(|w_j^{AB}| - \max_k \left\{w_{jk}^{AB}\right\}\right)}{S}, \tag{6}$$

where $k = 1 \ldots N^B$ and ν is a $BART$ template.

The error of a particular $BART^A$ template is (4) times (5). The research in this paper deals with concept learning. Thus, the $BART^B$ will have only two templates, one for 0 and one for 1. We initialize BARTMAP by allowing all input patterns to cluster into one $BART^A$ template. Note that because we use the most likely $BART^B$ template to label this cluster, the error will be no greater than 50% after one pass through the training data. Also, BARTMAP is operated in batch, or off-line, mode.

BARTMAP then proceeds to raise the vigilance of any $BART^A$ cluster template which contributes more than ε_d to the total error, where ε_d denotes the desired training error. If no $BART^A$ template contributes more than ε_d to the total error, and yet the total error is still greater than ε_d, then the vigilance is raised for all $BART^A$ cluster templates with maximum error contribution.

In this way BARTMAP proceeds from a very error prone single cluster to many clusters with small overall total error. The idea in boosting is to improve overall performance through multiple passes of the training data, and to focus in on those training examples which are in error. BARTMAP focuses on those templates where error is highest and splits them into smaller templates with less error, improving overall performance.

In BARTMAP, we want to allow some error into the labeled category formation during training. After seeing more training data this error will be reduced by boosting. The extra training data may just be repeated instances of data seen previously, and in this case BARTMAP's performance will increase on the training data. If the training data is properly representative of the underlying data distribution, then the completely boosted network will perform well.

BARTMAP will start in an "unlearned" state, as with ARTMAP. After one phase or epoch of learning, BARTMAP will have non-zero error, but it will not be greater than 50%. During each succeeding phase of learning, BARTMAP will attempt to reduce the error of the category which has highest error. A category's error will be reduced by increasing its vigilance parameter value. Initially all categories will be started with low vigilance values, meaning that each category can encompass data with a large spatial extent. As training proceeds, vigilance values for those categories with high error will be raised to reduce their extents designed to lower overall error. Notice, however, that a category that only contains a single data instance will necessarily have 0 error, but a category that has as few as two data instances can have as much as 50% error.

4. Empirical Results

For our empirical results, we compare the generalization performance of BARTMAP with ARTMAP as well as PROBART and ART-EMAP. First, each of the four learning networks were trained on data generated from three different 2D Gaussian distributions. These distributions consisted of two well-separated gaussians, two overlapping gaussians with the same means and different variances, and two overlapping gaussians with different means and the same variances. Next ARTMAP and BARTMAP are trained on steadily increasing size data sets and tested on many newly generated examples to estimate each network's generalization error percentage.

One of the 2D gaussians was labeled 0 and the other 1, to allow for concept learning. All data generated according to the distributions was normalized to fit within the unit square so that the Fuzzy ART architecture can be used.

In our experiment, each network type was trained on two sets of paired training and testing data samples. The first set contained 100 pairs, each consisting of 100 training samples paired with 1000 test samples drawn according to one of the three distributions. The second set contained 10 pairs of 1000 training samples paired with 10000 test samples again drawn according to one of the three distributions. The second set enabled us to see how each of the learning techniques scaled to larger sample sizes.

An ART^A vigilance of 0.9 and ART^B vigilance of 1.0 was used for ARTMAP, PROBART and ART-EMAP. A MAP field vigilance of 1.0 was also used for ARTMAP. A decision criterion (DC) value of 2.0 was used for ART-EMAP and $p = 10$ as well as $q = 3$ for contrast enhancement.

BARTMAP was run using 0.1 as a starting value for $BART^A$ vigilance values, and 0.1 was also used as a step size for increasing these values. A vigilance of 1.0 was used in BARTMAP for ART^B. BARTMAP was executed to a desired error tolerance of 0.1 for distribution 1, 0.2 for distribution 2, and 0.25 for distribution 3.

Well-Separated 2D Gaussians Distribution 1 was designed to ensure that ARTMAP, PROBART, ART-EMAP and BARTMAP were executing properly on well separated concept classes. One gaussian had mean $(5, 15)$ and the other $(15, 5)$ both with a variance of $(0.2, 0.2)$.

All four learning methods performed without error on the test data after learning the training data. BARTMAP did make a very small number of errors on test data in both the first and second sets. The errors did go down from the first set compared with the second set, however, indicating fewer outliers.

Overlapping Gaussians–Case 1 Our second experiment is a difficult problem where one 2D gaussian sits on top of the other one. Both 2D gaussians had mean $(10, 10)$, and one had a variance of $(0.5, 0.5)$ while the other had a variance of $(2.0, 2.0)$.

This problem does not have an errorless solution, and in fact the best separator, the quadratic where the two gaussians intersect, has a non-zero Bayes error.

Table 1: Distribution 2 - Set 1

technique	*epochs*	*templates*	*NP*	*errors*
ARTMAP	4.0	19.5	32.1	197.9
PROBART	1.0	11.7	67.2	611.4
ART-EMAP	4.1	18.2	36.5	473.4
BARTMAP	17.5	7.6	34.1	211.9

In table 1, we see the learning performance of ARTMAP, PROBART, ART-EMAP and BARTMAP on the problem at hand averaged over 100 sets each consisting of 100 training samples and 1000 test samples. The second column shows the average number of passes through the training data, i.e., epochs, needed to reach a solution. Notice that PROBART executes in 1 epoch, since subsequent passes through the same data do not alter the architecture. The third column gives the average number of F_2 templates necessary in the ART^A or $BART^A$ module. The fourth column lists the average number of no predictions reached on test data. A no prediction is output when an input pattern does not match, within vigilance, any of the F_2 templates. Thus, a no prediction is indicative of portions of the input space not covered by any F_2 template. Finally, the last column lists the average number of errors out of 1000 on the test data.

BARTMAP requires a considerable number of passes through the data, but its solution has the fewest number of F_2 templates while still maintaining nearly the same error percentage as ARTMAP.

Table 2: Distribution 2 - Set 2

technique	*epochs*	*templates*	*NP*	*errors*
ARTMAP	7.6	135.3	80.3	1890.5
PROBART	1.0	32.3	169.5	6347.8
ART-EMAP	9.8	120.6	5250.0	1064.2
BARTMAP	22.9	8.7	42.5	1794.0

In table 2, we see how the four learning techniques scale up to larger training/test sizes. Both ARTMAP and BARTMAP reduce the test set error percentage, and indeed both are below 20% error, but ARTMAP requires many more F_2 templates than BARTMAP, while BARTMAP requires more passes through the data.

Overlapping Gaussians–Case 2 The last pair of distributions considered again has no perfect, errorless solution. It consists of two overlapping 2D gaussians with different means. Thus, the two distributions overlap side-by-side, and a linear boundary is all that is necessary to properly separate them. One 2D gaussian had mean $(8, 12)$, and the other one had mean $(12, 8)$, while both had a variance of 2.0.

Table 3: Distribution 3 - Set 1

technique	*epochs*	*templates*	*NP*	*errors*
ARTMAP	3.5	31.9	23.6	346.2
PROBART	1.0	21.8	109.0	283.5
ART-EMAP	3.5	29.7	173.5	72.6
BARTMAP	9.6	3.7	13.5	305.1

As table 3 shows, all four learning architectures perform to nearly the same degree of accuracy. However, ARTMAP, PROBART and ART-EMAP have a higher degree of no predictions than BARTMAP, and BARTMAP requires only a small number of F_2 templates on average.

Table 4: Distribution 3 - Set 2

technique	*epochs*	*templates*	*NP*	*errors*
ARTMAP	7.1	210.8	22.1	3406.9
PROBART	1.0	56.8	153.8	5392.0
ART-EMAP	8.7	195.2	4868.9	103.3
BARTMAP	43.2	14.9	45.3	2670.1

In the second set of testing, table 4, we can see that for an order of magnitude more training data ARTMAP requires nearly an order of magnitude more F_2 templates than BARTMAP. PROBART has many more test set errors, an indication that it was not designed for classification. ART-EMAP has many more no predictions, indicating less test space is covered by its F_2 templates.

We continued to compare the generalization performance of BARTMAP with with Fuzzy ARTMAP by testing each of these neural network models as we increase the number of training samples. We estimate the generalization by taking a trained network and feeding it randomly generated examples until the error percentage remains within a fixed bound, i.e., 0.1%. Fuzzy ARTMAP and BARTMAP were tested on networks trained with 50, 100, 1000 and 10000 training samples for both distribution 2 and distribution 3. The numbers obtained were averaged over 10 networks trained using sets of each of the four sizes mentioned.

We expected to see a convergence of error percentage for both Fuzzy ARTMAP and BARTMAP towards the Bayes limit as the number of training samples increased. The actual Bayes error values have not been calculated at this time, but we do know that they are below the estimated values seen in our research. Note that the error percentage value, for both Fuzzy ARTMAP and BARTMAP, includes both wrongly classified instances as well as those instances that cannot be predicted.

Overlapping Gaussians–Case 1 Each trained network was tested on at least 30,000 randomly generated test samples. As the reader can see, ARTMAP has a better error percentage for training set sizes of 50 and 100, but BARTMAP improves to just over 10% better than Fuzzy ARTMAP at 10000 training samples.

Table 5: ARTMAP on Distribution 2

training size	*epochs*	*templates*	*% error*
50	3.0	12.5	0.252153
100	3.9	20.0	0.235387
1000	7.6	133.8	0.196986
10000	9.5	1250.5	0.189153

Table 6: BARTMAP on Distribution 2

training size	*epochs*	*templates*	*% error*
50	10.6	4.7	0.311179
100	12.4	6.3	0.272662
1000	16.4	11.0	0.188971
10000	19.2	8.3	0.179572

Overlapping Gaussians–Case 2 Each trained network was tested on at least 35,000 randomly generated test samples. With distribution 3, BARTMAP starts out slightly better than Fuzzy ARTMAP at 50 training samples but improves to 27%, quite a bit better than 35% for Fuzzy ARTMAP at 10000 training samples.

Table 7: ARTMAP on Distribution 3

training size	*epochs*	*templates*	*% error*
50	2.9	20.1	0.364948
100	3.7	32.9	0.373397
1000	7.2	207.4	0.351424
10000	9.6	2205.4	0.348173

Table 8: BARTMAP on Distribution 3

training size	*epochs*	*templates*	*% error*
50	10.1	4.1	0.324647
100	9.2	4.2	0.319004
1000	15.5	7.9	0.270913
10000	23.8	12.6	0.256323

The fact that both Fuzzy ARTMAP and BARTMAP produce networks with smaller error percentages as the number of training samples increases is not surprising. Both of these learning techniques will increase in performance as the number of training samples increases. Fuzzy ARTMAP performance increases at a slower rate than BARTMAP due to "overfitting". The experiments presented were designed to have a significant degree of overlap which induced "overfitting" with Fuzzy ARTMAP. Fuzzy ARTMAP also requires a tremendous number of F2 template nodes as compared to BARTMAP when the number of training samples increases.

5. Conclusions

After conducting the experiments, we have seen that BARTMAP is a reasonable alternative to ARTMAP, especially in learning situations where there is overlap between concept classes and no exact solution. Another benefit that BARTMAP provides is a reduction in the number of F_2 templates necessary for learning, at the expense of many repeated passes through the data. BARTMAP can execute similar to Fuzzy ARTMAP by requiring that it achieve 0 error. One of the reasons that BARTMAP is successful even at 20% desired error is that it does not overfit the training data as happens with Fuzzy ARTMAP. In fact, lowering the desired error actually leads to an increase in errors for BARTMAP beyond a certain point in problems with non-zero Bayes error.

At present, we are looking into ways of improving BARTMAP's performance and reducing the number of epochs needed for learning. We are also looking into applying BARTMAP on other concept problems as well as analyzing its generalization behavior theoretically.

Acknowledgments

This research was supported by a grant from The Boeing Company under contract W-300445.

References

[1] Robert E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[2] Gail A. Carpenter, Stephen Grossberg, and John H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, no. 5, pp. 565–588, 1991.

[3] Gail A. Carpenter, Stephen Grossberg, Natalya Markuzon, John H. Reynolds, and David B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698–713, 1992.

[4] Shaun Marriott and Robert F. Harrison, "A modified fuzzy ARTMAP architecture for the approximation of noisy mappings," *Neural Networks*, vol. 8, no. 4, pp. 619–641, 1995.

[5] Gail A. Carpenter and William D. Ross, "ART-EMAP: A neural network architecture for object recognition by evidence accumulation," *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 805–818, 1995.

[6] Chee Peng Lim and Robert F. Harrison, "A modified fuzzy ARTMAP approaches bayes optimal classification rates: An empirical demonstration," *Neural Networks*, vol. 9, no. 5, pp. 755–774, 1996.

[7] Michael Kearns and Yishay Mansour, "On the boosting ability of top-down decision tree learning algorithms," *ATT Technical Report*, 1995.