1999-05

Comparison of Classifiers for Radar Emitter Type Identification

Comparison of classifiers for radar emitter type identification

Eric Granger, Stephen Grossberg, Pierre Lavoie, and Mark Rubin

May, 1999

Technical Report CAS/CNS-99-014

Boston University Center for Adaptive Systems and
Department of Cognitive and Neural Systems
677 Beacon Street
Boston, MA 02215

Comparison of Classifiers for Radar Emitter Type Identification

Eric Granger[1,2], Stephen Grossberg[3], Pierre Lavoie[1] and Mark A. Rubin[3]

[1] Defence Research Establishment Ottawa
Department of National Defence, Ottawa, On., Canada, K1A 0Z4
{eric.granger, pierre.lavoie}@dreo.dnd.ca

[2] Department of Electrical and Computer Engineering
École Polytechnique de Montréal, Montreal, Qc., Canada, H3C 3A7
egranger@grm94.polymtl.ca

[3] Department of Cognitive and Neural Systems
Boston University, Boston, MA 02215, USA
{steveg, rubin}@cns.bu.edu

Abstract

ARTMAP neural network classifiers are considered for the identification of radar emitter types from their waveform parameters. These classifiers can represent radar emitter type classes with one or more prototypes, perform on-line incremental learning to account for novelty encountered in the field, and process radar pulse streams at high speed, making them attractive for real-time applications such as electronic support measures (ESM). The performance of four ARTMAP variants – ART-EMAP (Stage 1), ARTMAP-IC, fuzzy ARTMAP and Gaussian ARTMAP – is assessed with radar data gathered in the field. The k nearest neighbor (kNN) and radial basis function (RBF) classifiers are used for reference. Simulation results indicate that fuzzy ARTMAP and Gaussian ARTMAP achieve an average classification rate consistently higher than that of the other ARTMAP classifiers, and comparable to that of kNN and RBF. ART-EMAP, ARTMAP-IC and fuzzy ARTMAP require fewer training epochs than Gaussian ARTMAP and RBF, and substantially fewer prototype vectors (thus, smaller physical memory requirements and faster fielded performance) than Gaussian ARTMAP, RBF and kNN. Overall, fuzzy ARTMAP performs at least as well as the other classifiers in both accuracy and computational complexity, and better than each of them in at least one of these aspects of performance. Incorporation into fuzzy ARTMAP of the MT- feature of ARTMAP-IC is found to be essential for convergence during on-line training with this data set.

1 Introduction

Radar electronic support measures (ESM) involve the search for, interception, location, analysis and identification of radiated electromagnetic energy for military purposes. This provides valuable information for real-time threat detection, threat avoidance, and for timely deployment of counter-measures [1] [2] [3].

A critical function of radar ESM is the real-time identification of the radar type associated with each pulse train, or continuous wave signal, that is intercepted. This task is exceedingly challenging owing to increases in environment density (e.g., pulse Doppler radars can transmit hundreds of thousands of pulses per second); dynamically changing environments; multiplication and dispersion of the modes for military radars; agility in parameters like pulse repetition interval, radio frequency and scan; unknown and reserve modes for which no ESM library entry exists; overlap between the parameters of different radar types in the ESM library; and noise and propagation effects that lead to erroneous or incomplete signal characterization. New classification algorithms to perform radar type identification are thus sought.

In this paper, artificial neural networks are examined using computer simulations. Since artificial neural network classifiers are trained on samples of the data that they are intended to identify, radar data collected in the field were used in the simulations. From an ESM standpoint, training a system directly on radar data is a radical departure from current practice. At present, data are collected, analyzed, and distilled into ESM libraries off-line by skilled analysts. New libraries are disseminated to the field as needed. One inconvenience of the current approach is that it is time-consuming and does not allow rapid learning of new radar modes in the field. The neural network approach may thus constitute an interesting addition to radar ESM.

An attractive feature of ARTMAP [4] neural network classifiers is that new classes can be learned incrementally without retraining on the whole data set. ARTMAP networks can also learn one or more prototypes to represent a radar type class, and lend themselves well to high speed processing of radar pulses. Four networks based on ARTMAP are compared in this paper. They are ART-EMAP (Stage 1) [5], ARTMAP-IC [6], fuzzy ARTMAP [7] and Gaussian ARTMAP [8] [9]. The k nearest neighbor (kNN) [10] and radial basis function (RBF) [11] classifiers are used as non-parametric and parametric references, respectively. Parameters used for classification are radio frequency, pulse repetition interval, and pulse width. The performance of the classifiers is assessed in terms of classification rate and computational complexity.

The ARTMAP neural network classifier and the four variants in this study are briefly reviewed in the next section. The radar data set used to train and test the classifiers is presented in Section 3. Simulation results are then presented and analyzed in Section 4.

2 ARTMAP neural network classifiers

ARTMAP refers to a family of neural network architectures for self-organizing recognition and prediction [4] [7]. They perform incremental supervised learning of multi-dimensional mappings of an arbitrary sequence from the input space to a binary output space.

A simplified ARTMAP architecture is built by combining an ART [12] neural network with a map field. The ART network consists of two fully connected layers of nodes: an M' node input layer, $F1$, and an N node competitive layer, $F2$. A set of weights $\mathbf{W} = \{w_{ji} : j = 1, 2, ..., N; i = 1, 2, ..., M'\}$ is associated with the $F1 - F2$ layer connections. Each $F2$ node j represents a category in the input space, and stores a prototype vector $\mathbf{w}_j = (w_{j1}, w_{j2}, ..., w_{jM'})$. The $F2$ layer is connected, through associative links, to an L node map field F^{ab}, where L is the number of classes in the output space. A set of binary weights $\mathbf{W}^{ab} = \{w_{jk}^{ab} : j = 1, 2, ..., N; k = 1, 2, ..., L\}$ is associated with the $F2 - F^{ab}$ connections. The vector $\mathbf{w}_j^{ab} = (w_{j1}^{ab}, w_{j2}^{ab}, ..., w_{jL}^{ab})$ relates $F2$ node j to one of the L output classes. $F1$, $F2$ and F^{ab} activity patterns are denoted by $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{x}^{ab}$, respectively.

During training, an ARTMAP classifier learns the association between training set patterns $\mathbf{a} = (a_1, a_2, ..., a_M)$ (fed to the $F1$ layer) and respective supervision patterns $\mathbf{b} = (b_1, b_2, ..., b_L)$ (fed to the map field F^{ab}). The binary supervision patterns are coded to have unit value $b_K = 1$ correspond to the target class label K for $\mathbf{a}$, and zero elsewhere. The following algorithm describes the operation of an ARTMAP classifier in learning mode.

1. **Initialize weights and parameters:** Initially, all the neurons of $F2$ are uncommitted, all weight values w_{ji} are initialized to 1, and all weight values w_{jk}^{ab} are set to 0. Values of parameters such as the vigilance baseline $\bar{\rho} \in [0, 1]$ are set.

2. **Input pattern coding:** When a training pair $(\mathbf{a}, \mathbf{b})$ is presented to the network, $\mathbf{a}$ undergoes preprocessing, and yields pattern $\mathbf{A} = (A_1, A_2, ..., A_{M'})$. The vigilance parameter ρ is reset to its baseline value $\bar{\rho}$. The components of field activity patterns $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{x}^{ab}$ are reset to zero.

3. **Prototype selection:** Pattern $\mathbf{A}$ activates layer $F1$ ($\mathbf{x} = \mathbf{A}$) and is propagated through weighted connections $\mathbf{W}$ to layer $F2$. Each $F2$ node j activates according to the *choice function*. The $F2$ layer produces a binary, winner-take-all pattern of activity $\mathbf{y} = (y_1, y_2, ..., y_N)$, where only node $j = J$ with the greatest activation value remains active ($y_J = 1$). Node J propagates its prototype vector $\mathbf{w}_J$ onto $F1$ and the *vigilance test* is performed. This test compares the degree of match between $\mathbf{w}_J$ and $\mathbf{A}$ to a vigilance parameter $\rho \in [0, 1]$. If this test is satisfied, node J remains active and resonance is said to occur. Otherwise, the network inhibits the active $F2$ node and searches for another node J that passes the vigilance test. If such a node does not exist, an uncommitted $F2$ node becomes active and undergoes learning (go to Step 5).

4. **Class prediction:** Pattern $\mathbf{b}$ is fed to the map field, while the $F2$ activity pattern $\mathbf{y}$ is propagated via associative connections $\mathbf{W}^{ab}$ to the map field F^{ab}. The latter input activates F^{ab} nodes according to a *prediction function* and the most active F^{ab} node K yields the class prediction ($K = k(J)$). If node K constitutes an incorrect class prediction, a *match tracking* signal raises vigilance just enough to induce another search among $F2$ nodes (Step 3). This search continues until either an uncommitted $F2$ node becomes active (learning ensues at Step 5), or a node J that has previously learned the correct class prediction K becomes active.

5. **Learning:** Prototype vector $\mathbf{w}_J$ is updated, and, if J corresponds to a newly-committed node, a permanent associative link to F^{ab} is created. A new association between $F2$ node J and F^{ab} node K ($k(J) = K$) is learned by setting $w^{ab}_{Jk} = 1$ for $k = K$, where K is the target class label for $\mathbf{a}$.

Once the weights ($\mathbf{W}$ and $\mathbf{W}^{ab}$) have converged for the training set patterns, ARTMAP can predict a class label for an input pattern by performing Steps 2, 3 and 4 without any testing. A pattern $\mathbf{a}$ that activates node J is predicted to belong to the class $K = k(J)$.

Although the first ARTMAP classifier [4] is limited to processing binary-valued input patters, the ART-EMAP, ARTMAP-IC, fuzzy ARTMAP (FAM) and Gaussian ARTMAP (GAM) classifiers can process both analog and binary-valued input patterns. The rest of this section highlights the main differences between these four ARTMAP variants. (Refer to Table 1 for algorithmic differences.)

2.1 ART-EMAP, ARTMAP-IC and fuzzy ARTMAP

ART-EMAP, ARTMAP-IC and FAM variants employ fuzzy ART [13] as the ART network. With fuzzy ART, a transformation called complement coding doubles the number of components in the input pattern ($M' = 2M$), which becomes $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c) = (a_1, a_2, ..., a_M; a^c_1, a^c_2, ..., a^c_M)$, where $a^c_i = (1 - a_i)$. With complement coding and fast learning ($\beta = 1$), fuzzy ART represents category j as hyperrectangle R_j that just encloses all the training set patterns $\mathbf{a}$ to which it has been assigned. That is, a $2M$-dimensional prototype vector $\mathbf{w}_j = (w_{j1}, w_{j2}, ..., w_{j2M})$ records the largest and smallest component values of training set patterns $\mathbf{a}$ placed in the j^{th} category.

ART-EMAP (Stage 1) and ARTMAP-IC are extensions of FAM that produce a binary winner-take-all pattern $\mathbf{y}$ when training, but use distributed activation of coded $F2$ nodes when testing. ARTMAP-IC is further extended in two ways. First, it biases distributed test set predictions according to the number of times $F2$ nodes are assigned to training set patterns. Second, it uses negative match tracking (MT-) (i.e., negative ϵ values) to address the problem of inconsistent cases, whereby identical training set patterns correspond to different classes labels.

2.2 Gaussian ARTMAP

GAM differs significantly from the other three. It represents category j as a Gaussian density function, defined by two vectors: its mean $\mu_j = (\mu_{j1}, \mu_{j2}, ..., \mu_{jM})$ and its standard deviation $\sigma_j = (\sigma_{j1}, \sigma_{j2}, ..., \sigma_{jM})$ (they replace prototype vector $\mathbf{w}_j$). A scalar, n_j, accumulates the amount of relative activation obtained by $F2$ node j on training set patterns.

During training, the number of committed $F2$ nodes, N_c, is initially set to 0. Newly-committed $F2$ nodes increment N_c, and undergo the initialization step (instead of the learning step). The committed $F2$ nodes that pass the vigilance test for pattern $\mathbf{a}$ are allowed to activate, and distribute a pattern of activity

$\mathbf{y} = (y_1, y_2, ..., y_{N_c})$. Match tracking and learning are performed according to the relative activation over the "ensemble" E_K of $F2$ nodes linked to the predicted F^{ab} node K. The relative activation over E_K is defined by the distributed pattern $\mathbf{y}^* = (y_1^*, y_2^*, ..., y_{N_c}^*)$, where $y_j^* = y_j / \sum_{l \in E_K} y_l$ only if $j \in E_K$, and $y_j^* = 0$ otherwise.

Table 1: Distinctive equations used by the ARTMAP networks. With ART-EMAP, ARTMAP-IC and FAM, $|\cdot|$ is the norm operator ($|\mathbf{w}_j| \equiv \sum_{i=1}^{2M} |x_i|$), $\wedge$ is the fuzzy AND operator ($(\mathbf{A} \wedge \mathbf{w}_j)_i \equiv \min(A_i, w_{ji})$), α is the *choice parameter*, and β is the *learning rate parameter*. With GAM, γ is the initial standard deviation assigned to newly-committed $F2$ nodes.

Algorithmic step	ARTMAP classifier	
	ART-EMAP, ARTMAP-IC & FAM	GAM
1.Initialization:	$\alpha > 0,\ \beta \in [0, 1]$	$\mu_J = \mathbf{A},\ \sigma_{Ji} = \gamma\ (\gamma > 0),\ w_{JK}^{ab} = 1,\ n_J = 1$
2.Input coding:	$\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)\ (M' = 2M)$	$\mathbf{A} = \mathbf{a}\ (M' = M)$
3.Prototype selection:		
– choice function	$T_j(\mathbf{A}) = \lvert\mathbf{A} \wedge \mathbf{w}_j\rvert / (\alpha + \lvert\mathbf{w}_j\rvert)$	$g_j(\mathbf{A}) = \begin{cases} \frac{n_j}{\prod_{i=1}^{M} \sigma_{ji}} G_j(\mathbf{A}) & \text{if } G_j(\mathbf{A}) > \rho \\ 0 & \text{otherwise} \end{cases}$
– vigilance test	$\lvert\mathbf{A} \wedge \mathbf{w}_J\rvert \geq \rho M$	$G_j(\mathbf{A}) = \exp\left\{-\frac{1}{2} \sum_{i=1}^{M} \frac{(A_i - \mu_{ji})^2}{\sigma_{ji}^2}\right\} > \rho$
– $F2$ activation	$y_j = 1$ only if $j = J$	$y_j = g_j / (0.01 + \sum_{l=1}^{N_c} g_l)$
4.Class prediction:		
– prediction function	$S_k^{ab}(\mathbf{y}) = \sum_{j=1}^{N} y_j w_{jk}^{ab}$	$S_k^{ab}(\mathbf{y}) = \sum_{j=1}^{N_c} y_j w_{jk}^{ab}$
– match track	$\rho' = (\lvert\mathbf{A} \wedge \mathbf{w}_J\rvert / M) + \epsilon$	$\rho' = \exp\left\{-\frac{1}{2} \sum_{j \in E_K} y_j^* \sum_{i=1}^{M} \frac{(A_i - \mu_{ji})^2}{\sigma_{ji}^2}\right\} + \epsilon$
5.Learning:		
– prototype update	$\mathbf{w}'_J = \beta(\mathbf{A} \wedge \mathbf{w}_J) + (1 - \beta)\mathbf{w}_J$	$n'_j = n_j + y_j^*$ $\mu'_{ji} = (1 - \frac{y_j^*}{n_j})\mu_{ji} + \frac{y_j^*}{n_j} A_i$ $\sigma'_{ji} = \sqrt{(1 - \frac{y_j^*}{n_j})\sigma_{ji}^2 + \frac{y_j^*}{n_j}(A_i - \mu_{ji})^2}$

3 Radar pulse data

The data set used to evaluate the classifiers contains approximately 100,000 consecutive radar pulses gathered over 16 seconds during a field trial by the Defense Research Establishment Ottawa. After the trial, an ESM analyst manually separated trains of pulses coming from different emitters. Each pulse was then tagged with two numbers: a radar type number and a mode number. A single type of radar can use several modes to perform various functions. Since ESM trials are complex and never totally controlled, not all pulses can be tagged and a residue is obtained. Residue pulses were discarded for this study.

The patterns used for this study contain 3 parameters: radio frequency (RF), pulse repetition interval (PRI) and pulse width (PW). The RF and PW parameters are measured on each individual pulse, whereas PRI is derived from the time-of-arrival (TOA) of pulses from a single emitter. For simplicity, a TOA deinterleaver was assumed to be capable of correctly sorting the n_k pulses belonging to each active emitter mode k, and then computing: $\text{PRI}_k(i) = \text{TOA}_k(i) - \text{TOA}_k(i-1)$ for $i = 2, 3, ..., n_k$. Note that the first pattern from each active emitter mode is omitted from the comparison. Also, due to the circular scanning action of some radar emitters, pulses are recorded in bursts. The first pulse of each scan (or burst) is also omitted. Finally, the parameters were linearly normalized so that $a_i \in [0, 1]$, $i = 1, 2, 3$.

Once deinterleaved and tagged, the data used to train and test the classifiers contain 52,192 radar pulses from 34 modes, each one belonging to one of 15 different radar types. The data feature bursts of high pulse densities, multiple emitters of a same type, modes with overlapping parametric ranges, radars transmitting at different pulse rates, and emitters switching modes. The sophistication of the radar types range from simple (constant RF and PRI) to fairly complex (pulse-to-pulse RF and PRI agility). The data also contain direction of arrival (DOA), but this parameter is not used here.

4 Simulation results and discussion

Matlab code was written for all the classifiers. Prior to each simulation trial, the data set described in Section 3 was partitioned into training and test subsets. 50% of the data from each radar type was selected at random to form the training subset. Then, training subset patterns $\mathbf{a}$ were repeatedly presented to each classifier along with their supervision patterns $\mathbf{b}$ (coding radar types) until convergence was reached – when the sum-squared-error (SSE) of prototype weights was less that 0.001 for two successive epochs. An epoch is defined as a presentation of the training subset to a classifier in a TOA sequence. Finally, the test subset (the complete set less the training data) was presented to the trained classifiers for prediction.

The performance of the classifiers was compared in terms of the amount of resources they require during training, and their predictive accuracy on the test subset. The amount of resources required while training is measured in the 3 following ways. *Compression* refers to the average ratio of training patterns to committed F2 node. *Memory* is the number of normalized registers[1] needed to store the set of learned prototype vectors. The *convergence time* is the number of epochs required for the classifier to converge (SSE < 0.001). Predictive accuracy is measured with the *classification rate* – the ratio of correctly classified patterns over all test patterns.

Table 2 presents the average results (with corresponding standard deviations) of each classifier, from 20 independent simulation trials. The parameter settings were selected to achieve the best classification rate for the least amount of memory and convergence time during training.

Results in Table 2 indicate that FAM and GAM yield the highest average classification rates for the data set, followed by ARTMAP-IC and ART-EMAP. These classification rates are comparable to those obtained with the reference kNN and RBF classifiers. ART-EMAP, ARTMAP-IC and FAM attain their classification rates with greater compression (thus less memory requirement to store prototype vectors) than the other classifiers, and significantly fewer training epochs to converge than GAM and RBF.

4.1 Training set convergence

A convergence problem occurs with ART-EMAP, FAM and GAM whenever the training subset contains identical patterns (i.e., pulses in the same resolution cell) that belong to different radar types. The problem is aggravated since ARTMAP tends to segment the overlapping/scattered parts of classes into several very tiny, often minimum-sized prototypes. The effect is a proliferation of identical prototypes for certain training set patterns.

Consider the following example. Assume that on the first training epoch, FAM learns two completely overlapping, minimum-sized prototypes, $\mathbf{w}_{A.1}$ (linked to class A) and $\mathbf{w}_{B.1}$ (linked to class B), for two identical pulse patterns, $\mathbf{a}_1$ and $\mathbf{a}_2$. In a subsequent epoch, $\mathbf{w}_{A.1}$ is initially selected to learn $\mathbf{a}_2$, since $T_{A.1} = T_{B.1} \simeq 1$, and $\mathbf{w}_{A.1}$ was created prior to $\mathbf{w}_{B.1}$ (index $A.1$ is smaller than $B.1$). Since $\mathbf{w}_{A.1}$ is not linked to class B, mismatch reset raises the vigilance parameter ρ to $(|\mathbf{A}_2 \wedge \mathbf{w}_{A.1}|/M) + \epsilon$, where $|\mathbf{A}_2 \wedge \mathbf{w}_{A.1}| = |\mathbf{A}_2 \wedge \mathbf{w}_{B.1}|$. As a result, $\mathbf{w}_{B.1}$ can no longer pass the vigilance test required to become selected for $\mathbf{a}_2$, and FAM must create another minimum-sized prototype $\mathbf{w}_{B.2} = \mathbf{w}_{B.1}$. From epoch to epoch, the same phenomenon repeats itself, yielding ever more prototypes $\mathbf{w}_{B.\mathrm{n}} = \mathbf{w}_{B.1}$ for n $= 3, 4, ..., \infty$.

ART-EMAP, FAM and GAM failed to converge while training on the radar data set. Results in Table 2 were obtained through manual termination by: (1) detecting, from epoch to epoch, the repeated creation of identical prototypes for the same training pulses, (2) pruning non-unique prototypes from memory, and (3) defining the convergence time as the number of epochs leading to the creation of non-duplicate prototypes only. Clearly this approach is cumbersome, and a better way of handling the problem was examined.

ARTMAP-IC was capable of converging incrementally on-line for the radar data. The feature of ARTMAP-IC that allows it to avoid the convergence problem is MT-; that is, the use of a negative ϵ

[1] A normalized register is a fixed-size register, with the number of bits equal to that needed to store the classifier's real values such as a_i, w_{ji}, ρ, etc.

Table 2: Average classification results for the radar data. The * indicates that the classifier was unable to converge for the training set on each trial (refer to Subsection 4.1).

Classifier	Evaluation criteria (std. deviation)			
	Accuracy	Resources		
	Classification rate	**Compression**	**Memory**	**Conv. time**
kNN ($k = 1$, $d_{\text{cityblock}}$)	0.996 (0.001)	1 (0)	80311 (0)	N/A
kNN ($k = 1$, $d_{\text{Euclidean}}$)	0.996 (0.001)	1 (0)	80311 (0)	N/A
RBF (spread = 0.05)	0.997 (0.001)	4.1 (0.1)	6367.9 (65.1)	2123.6 (21.7)
ART-EMAP* (Stage 1) ($\epsilon = 10^{-4}$, $\alpha = .001$, $\beta = 1$, $\bar{\rho} = 0$)	0.784 (0.074)	216.0 (27.1)	735.3 (89.4)	stopped at 3.9 (0.4)
ARTMAP-IC (complete) ($\epsilon = 10^{-4}$, $\alpha = .001$, $\beta = 1$, $\bar{\rho} = 0$)	0.840 (0.039)	217.2 (20.5)	727.5 (66.4)	3.8 (0.4)
FAM* ($\epsilon = 10^{-4}$, $\alpha = .001$, $\beta = 1$, $\bar{\rho} = 0$)	0.996 (0.001)	216.0 (27.1)	735.3 (89.4)	stopped at 3.9 (0.4)
GAM* ($\epsilon = 10^{-3}$, $\gamma = .0025$, $\bar{\rho} = 0$)	0.997 (0.001)	102.1 (4.3)	1536.0 (65.8)	stopped at 6.0 (1.1)
FAM with MT- ($\epsilon = 10^{-6}$, $\alpha = .001$, $\beta = 1$, $\bar{\rho} = 0$)	0.996 (0.001)	211.8 (32.1)	751.8 (90.6))	3.7 (0.5)
FAM with MT- and limited IC ($\epsilon = 10^{-4}$, $\alpha = .001$, $\beta = 1$, $\bar{\rho} = 0$)	0.996 (0.001)	217.0 (22.5)	727.5 (66.4)	3.8 (0.4)

value [6]. In the example above, mismatch reset would raise ρ but $\mathbf{w}_{B.1}$ would still pass the vigilance test. This allows ARTMAP-IC to learn fully overlapping prototypes for training set patterns that belong to different classes. As is seen from Table 2, and as is discussed further in Subsection 4.4, the complete ARTMAP-IC algorithm – MT-, distributed activation and instance counting – performs comparatively poorly on this data. This suggests that the best approach is ARTMAP-IC with MT- only, or, equivalently, FAM plus MT-.

4.2 Distributing and biasing test set activation

Simulation trials showed that the Q-max rule [6] for distributing $F2$ layer activation in ART-EMAP and ARTMAP-IC classifiers gives better results than both power or threshold rules [5] with the radar data. However, a priori choice of the Q parameter for this rule is critical for performance. The following choice of Q was found to give good results for the radar data: $Q = \min\{\lceil N_c/2L \rceil, 2L\}$, where L is the number of classes (15 in this study) and N_c is the number of committed $F2$ nodes. In particular, bounding Q to the low maximal value of $2L$ appears to reduce wild performance fluctuations.

Regardless of distributed activation, FAM performs better than its two extensions, ART-EMAP and ARTMAP-IC, on the radar data. Careful observation revealed that distributed activation for the test patterns makes more prediction errors because some radar type classes in the data set are very dispersed, are fragmented, and overlap one another. This is incompatible with class predictions that are based on the distribution of strongly activated $F2$ nodes among radar type classes (rather than on the most active $F2$ node). Indeed, an F^{ab} class node k that receives a very strong activation from one $F2$ node may have weaker overall activation than a F^{ab} class node h that receives a moderately strong activation from several $F2$ nodes. Perhaps this explains why kNN gives its best performance for $k = 1$, and degrades slowly as k grows. (For instance, its classification rate is 0.992 for $k = 9$ and $d_{\text{cityblock}}$.)

GAM involves training and testing with a distributed pattern of activity. $F2$ layer activation is distributed among nodes that pass the vigilance test. When training, each category $j \in E_K$ learns according to its relative activation for $\mathbf{a}$. $F2$ nodes learn a Gaussian mixture model of the input space. Although computationally intensive, the m-learning strategy allows GAM to achieve very high classification rates.

ARTMAP-IC and GAM accumulate weighting factors that depend on the quantity of training subset patterns assigned to each $F2$ node. This frequency information is used to bias predictions towards classes assigned the most training patterns. This constitutes an issue in radar ESM since some critical radars transmit very few pulses, while others transmit hundreds of thousands of pulses per second. Biasing prototype choices according to patterns in the time of arrival of pulses would be more appropriate.

4.3 Prototype representations

Given the quantization of parameter measurements, intercepted radar pulses fall into resolution cells. Since the measurement uncertainty of the three parameters used (RF, PRI and PW) are independent of each other, the radar type definitions are essentially rectangular. ART-EMAP, ARTMAP-IC and FAM use hyperrectangles to represent prototypes in the input space, and appear to be a better match for this type of data. GAM and RBF, on the other hand, represent prototypes with Gaussian density functions. This results in substantial segmentation of radar type classes, and in low compression for these two classifiers.

4.4 Enhancements to fuzzy ARTMAP

The bottom of Table 2 shows the results of two modified versions of FAM. The first version augments FAM with the MT- component of ARTMAP-IC. This results in a version of FAM that learns non-unique prototypes linked to different classes (thus converges for the radar data). As shown in Table 2, FAM with MT- performs as well, to within standard error, as kNN, with much less use of memory and without the convergence problem. As pointed out in [6] MT- is a better algorithmic approximation to the continuous-time version of the FAM neural network.

Figure 1 shows the average misclassification rate for FAM with MT- over the 16 second time span of the radar data. Each point on the figure represents the number of classification errors divided by the pulse count for the previous 0.25 seconds. The misclassification rate appears to be fluctuating, exceeding 2% on many occasions, and reaching a maximum value at 6%.

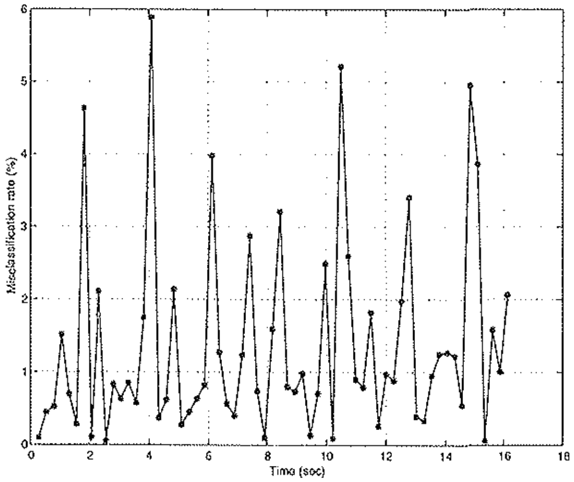

Figure 1: Average misclassification rate of FAM with MT- over time.

Although weighting predictions according to the number of training set patterns is not appropriate in our context, it may be useful on other data sets for FAM with MT- to produce better than a random guess for test patterns that activate the same $F2$ node[2]. This could result from using instance-weighted outputs only in the case where winning nodes are "inconsistent-case siblings," since in this case the only basis on which to choose one of the winning nodes over another may be the frequency with which they were winning nodes during training. Table 2 presents the results for FAM with MT- and "limited instance counting." With this variant, instances are still counted for all $F2$ nodes. However, it distributes activation weighted by the instance counts if and only if nodes J are a set that code for the same test pattern but map to different classes. Limited IC does not harm accuracy on the radar data set.

5 Conclusion

This paper presents a comparison of four neural network classifiers - ART-EMAP (Stage 1), ARTMAP-IC, fuzzy ARTMAP and Gaussian ARTMAP for the identification of radar emitter types associated with intercepted radar pulse trains. Their performance is measured in terms of resource allocation and accuracy.

In computer simulations using a radar data set collected in the field, FAM with negative match tracking (MT-) has performed consistently well. It gives one of the best classification rates, yet requires among the least amount of resources (shortest convergence time and least storage for prototypes) among ARTMAP classifiers. The MT- feature allows it to converge on-line despite the presence of training set categories that overlap completely yet belong to different classes.

6 Acknowledgements

This research was supported in part by the Defense Advanced Research Projects Agency and the Office of Naval Research ONR N00014-95-1-0409 (S. G. and M. A. R.), the National Science Foundation NSF IRI-97-20333 (S. G.), the Natural Sciences and Engineering Research Council of Canada (E. G.), and the Office of Naval Research ONR N00014-95-1-0657 (S. G.).

References

[1] Davies, C. L., Hollands, P., "Automatic Processing for ESM," *Proc. IEE*, **129:3** **(F)**, 164-171, June 1982.

[2] Sciortino, J. C., "Autonomous ESM Systems," *Naval Engineers Journal*, 73-84, November 1997.

[3] Wiley, R. G., *Electronic Intelligence: The Analysis of Radar Signals*, 2nd edition, Artech House, 1993.

[4] Carpenter, G. A., Grossberg, S., Reynolds, J. H., "ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network," *Neural Networks*, **4**, 565-588, 1991.

[5] Carpenter, G. A., Ross, W. D., "ART-EMAP: A Neural Network Architecture for Object Recognition by Evidence Accumulation," *IEEE Transactions on Neural Networks*, **6**, 805-818, 1995.

[6] Carpenter, G. A., Markuzon, N., "ARTMAP-IC and Medical Diagnosis: Instance Counting and Inconsistent Cases," *Neural Networks*, **11**, 323-336, 1998.

[7] Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., Rosen, D. B., "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps," *IEEE Transactions on Neural Networks*, **3:5**, 698-713, 1992.

[8] Williamson, J. R., "Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps," *Neural Networks*, **9:5**, 881-897, 1996.

[9] Williamson, J. R., "A Constructive, Incremental-Learning Neural Network for Mixture Modeling and Classification," *Neural Computation*, **9**, 1517-1543, 1997.

[10] Cover, T. M., Hart, P. E., "Nearest Neighbor Patterns Classification," *IEEE Transactions on Information Theory*, **13**, 21-27, 1967.

[11] Chen, S., Cowan, C. F. N., Grant, P. M., "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, **2**, 302-309, 1991.

[12] Carpenter, G. A., Grossberg, S., "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer, Vision, Graphics and Image Processing*, **37**, 54-115, 1987.

[13] Carpenter, G. A., Grossberg, S., Rosen, D. B., "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System," *Neural Networks*, **4**, 759-771, 1991.

[2] Without limited IC, FAM with MT- breaks ties by choosing the $F2$ node with the smallest index.