Classification of Incomplete Data Using the Fuzzy ARTMAP Neural Network

Eric Granger[1,2], Mark A. Rubin[3,4], Stephen Grossberg[3] and Pierre Lavoie[1]

[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

The fuzzy ARTMAP neural network is used to classify data that is incomplete in one or more ways. These include a limited number of training cases, missing components, missing class labels, and missing classes. Modifications for dealing with such incomplete data are introduced, and performance is assessed on an emitter identification task using a database of radar pulses.

1 A taxonomy of data incompleteness

Data presented to a classifier, during either the training or testing phases, may be incomplete in one or more ways:

1. Limited number of training cases: It is of interest to know how the performance of the classifier declines as the amount of training data is decreased, so that, *e.g.*, more training data may be gathered, if necessary, before the classifier is put to use.

2. Missing components of the input patterns: For example, the information in the different components of the input patterns may come from different sensors, one or more of which may be temporarily unavailable.

3. Missing class labels during training: Some of the training data may have missing class labels. This is referred to as "semi-supervised learning" (Demiriz *et al.*, 1999) or "partially supervised clustering" (Bensaid *et al.*, 1996). ("Missing class labels during testing" is, of course, just the usual situation.)

4. Missing classes: Some classes that were not present in the training set may be encountered during testing. When the classifier encounters a pattern belonging to such an unfamiliar class, it should "flag" the pattern as belonging to an unfamiliar class, rather than make a meaningless guess as to its identity. This may be implemented by using "familiarity discrimination" (Carpenter *et al.*, 1997).

(a) Pure familiarity discrimination. As is common practice when evaluating the performance of a classifier, the classifier does not learn during the testing phase. Test patterns which are flagged as unfamiliar are not processed further. In addition to high accuracy of classification of familiar patterns, the quality of the classifier is measured by a high "hit rate"—fraction of familiar-class test patterns correctly declared to belong to classes familiar during testing and classified (correctly or not)—and low "false alarm rate"—fraction of unfamiliar-class test patterns incorrectly declared familiar by the classifier.

(b) Learning of unfamiliar classes (LUC). The classifier continues to learn during testing. When an unfamiliar class is flagged, the classifier defines a new class, and the criteria for familiarity discrimination are adjusted as necessary. Subsequent test patterns may be declared by the classifier to be "familiar" and

[4]Address after March 1, 2000: Sensor Exploitation Group, MIT Lincoln Laboratory, 244 Wood St, Lexington, MA 02420

classified as belonging either to classes encountered during training *or* to the "newly-minted" classes; or they may be declared to be "unfamiliar," in which case another new class will be defined. (The normal adjustment of weight values during learning is also allowed during this phase.) The false-alarm rate for an LUC classifier is the fraction of unfamiliar-class (*i.e.*, not encountered during the training phase) test patterns not either flagged as unfamiliar or assigned to a "new" node defined during testing. An additional figure of merit for an LUC classifier is a "purity measure" such as the Rand score (Hubert and Arabie, 1985), which rewards the classifier for assigning test patterns belonging to different unfamiliar classes to different classes defined during testing, while penalizing it for creating too large a number of new classes during testing.

In this paper we present methods for dealing with the above types of incomplete data using the fuzzy ARTMAP neural network (Carpenter *et al.*, 1992) for classification. These methods are tested on a radar pulse data set that is described in Section 2. The details of the methods, and the results of their application to the radar pulse data, are described in Sections 3-6.

2 Radar pulse data

The data set used consists of approximately 100,000 consecutive radar pulses gathered over 16 seconds during a field trial by the Defense Research Establishment Ottawa. Each of these pulses was produced by one of fifteen different radar types. After the trial, an ESM (electronic support measures) analyst manually separated trains of pulses coming from different emitters. Each pulse j was then assigned a class label $C_j \in 1, \ldots, 15$, corresponding to the emitter type from which the analyst determined it to have come.[1] Since ESM trials are complex and never totally controlled, not all pulses can be tagged and a residue is obtained. Residual pulses were discarded for this study.

The input pattern $\mathbf{a}_j$ corresponding to the j^{th} pulse has three components: $\mathbf{a}_j = (RF_j, PW_j, PRI_j)$. RF is the radio frequency of the carrier wave, PW is the pulse width (temporal extent of the pulse), and PRI is the pulse repetition interval. The RF and PW components are by their nature associated with each individual pulse, whereas PRI is derived from the time-of-arrival (TOA) of pulses from a single emitter. For simplicity, we assume that, as part of the preprocessing, a TOA deinterleaver (Wiley, 1993) has correctly sorted the N_k pulses belonging to each active emitter type k, $k = 1, \ldots, 15$, and has computed, for each pulse j, $PRI_j = TOA_j - TOA_{j'}$, where j' is the pulse immediately preceding pulse j in the train of pulses coming from the emitter which produced pulse j.

Note that the first pulse from each 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 components of $\mathbf{a}_j$ were rescaled so that $a_{ji} \in [0, 1]$. This is required for the application of fuzzy ARTMAP. Once deinterleaved and tagged, the data set used to train and test the classifier contains 52,192 radar pulses from 34 modes, each one belonging to one of the 15 different radar types. The data feature bursts of high pulse densities, multiple emitters of the 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) information, but this is not used here.

3 Limited number of training cases

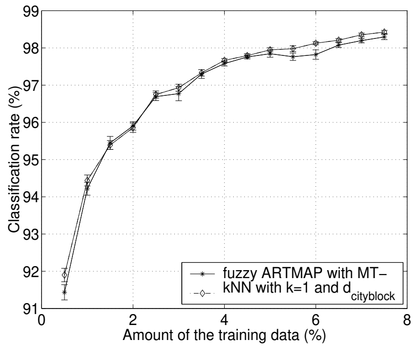
To avoid the problem of node proliferation that can arise when identical or nearly-identical input patterns in the training data correspond to different classes, a fuzzy ARTMAP variant termed MT- (Carpenter and Markuzon, 1998) is employed throughout this paper. After an incorrect prediction during training, the vigilance parameter is raised just enough to induce a search for another internal cluster, then lowered by a small amount $\epsilon > 0$. Simulations have indicated (Granger *et al.* 1999a) that, compared to several other variants of ARTMAP, as well as radial basis function and k-nearest neighbor (kNN) classifiers, this algorithm provides the most effective classification of the present data set in terms of accuracy and computational complexity (compression and convergence time).

The radar pulse data set was partitioned into training and test sets. 50% of the data from each radar type was selected at random to form the training set. Then, training set patterns $\mathbf{a}_j$ were repeatedly

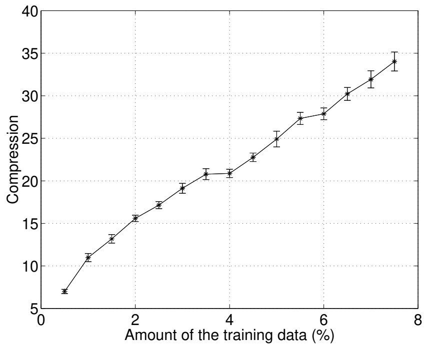
[1] A mode number was also assigned to each pulse. A single type of radar can use several modes to perform various functions. We do not here attempt to classify the pulses according to mode, so this label will be ignored.

presented, in order of TOA, to each classifier along with their class labels C_j until convergence was reached; that is, when the sum-squared-fractional-change (SSFC) of prototype weights was less than 0.001 for two successive epochs. An epoch is defined as a presentation of the training set to a classifier in a TOA sequence. Finally, the test set (the complete data set less the training data) was presented to the trained classifier for prediction. The results presented are averages over 20 random selections of the data to be used for training. Error bars are standard errors. The kNN classifier is shown for comparison.

Fig. 1(a) shows the effect on classification accuracy of reducing the amount of training data. Even when only 0.5% of the training data (about 130 pulses) is used, accuracy on the independent test set is 91.4%, compared to 99.6% when all the data is used. The notion that additional training examples beyond a certain point become "redundant" is borne out by Fig. 1(b), which shows compression increasing significantly as the number of training patterns is increased. (Compression refers to the ratio of training patterns to hidden layer nodes, a measure of efficiency of information storage.)



(a)



(b)

Figure 1: Limited number of training cases. (a) Fuzzy ARTMAP and kNN ($k = 1$) accuracy. (b) Fuzzy ARTMAP compression.

4 Missing components of the input patterns

Three strategies for addressing missing input components in fuzzy ARTMAP are:

1. Replacement by "0:" This strategy has been employed as part of the testing phase of Incremental ART (Aguilar and Ross, 1994). Input patterns $\mathbf{a}$ are fed to an $F1$ layer that implements *partial feature vector complement coding*, which allows for recognition based on a feature vector $\mathbf{A}'$. This vector has the usual complement-coded form, $\mathbf{A}' = (\mathbf{a}, \mathbf{a}^c)$, except that both the "on" component a_i and the corresponding complement-coded "off" component a_i^c of $\mathbf{A}'$ are set equal to 0 when the i^{th} component of the input pattern $\mathbf{a}$ is missing. We extended this approach to include the learning phase (see Table 1).

2. Replacement by "1:" Alternatively, both the on and off components of the complement-coded input pattern $\mathbf{A}'$ can be set equal to 1 when the i^{th} component is absent. With this strategy, as $|\mathbf{A}'|$ grows, the vigilance test $|\mathbf{w}_j \wedge \mathbf{A}'|/|\mathbf{A}'| > \rho$ becomes harder to pass. To compensate, the denominator $|\mathbf{A}'|$ is replaced by a fixed value M (the same value the complement-coded pattern has $\mathbf{A}'$ has in the absence of missing components).

3. Indicator vector: An indicator vector (Little and Rubin, 1987) $\delta = (\delta_1, \delta_2, ..., \delta_{2M})$ informs the fuzzy ARTMAP network as to the absence or presence of each component of an input pattern: $\delta_i = 1$ if component i is present, $\delta_i = 0$ if component i is missing, for $i = 1, \ldots M$, with $\delta_i \equiv \delta_{i-M}$ for $i = M+1, \ldots, 2M$. This strategy, unlike the other two, modifies the weight vector as well as the input vector in response to missing components.

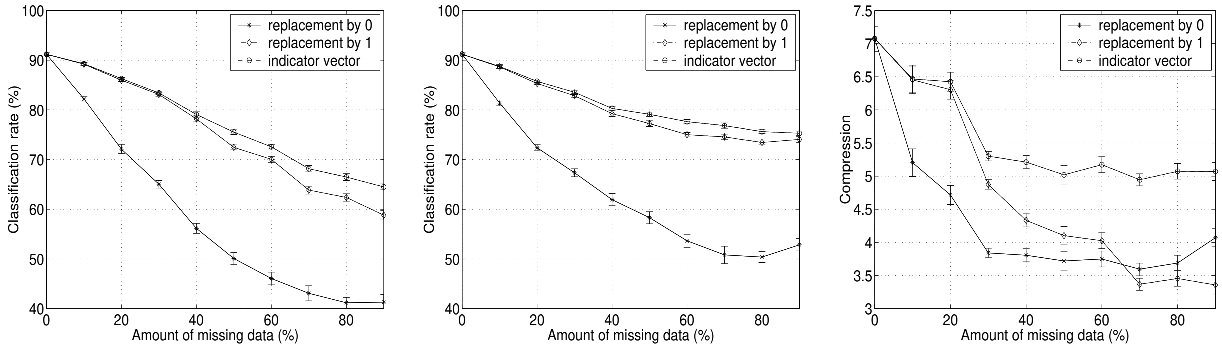
Table 1 summarizes the operation of these three strategies (for notational convenience, the indicator vector δ appears in the learning rules for all 3 strategies).

Training was performed with 0.5% of the available training data. A percentage of the components of either the training or test vectors from each emitter type were randomly chosen to be "missing" (although, if a particular choice of missing components would have left the vector with *no* components, another

Strategy	Input pattern coding	Prototype choice	Vigilance test	Prototype learning
Replacement by "0"	$\mathbf{A}' = (\mathbf{a}, \mathbf{a}^c)$; *BUT*, set $a_i = a_i^c = 0$ if i missing	$T_j(\mathbf{A}') = \frac{\lvert \mathbf{w}_j \wedge \mathbf{A}' \rvert}{\alpha + \lvert \mathbf{w}_j \rvert}$	$\frac{\lvert \mathbf{w}_j \wedge \mathbf{A}' \rvert}{\lvert \mathbf{A}' \rvert} \geq \rho$	$\mathbf{w}'_J = \delta\beta(\mathbf{A}' \wedge \mathbf{w}_J) + (1-\delta\beta)\mathbf{w}_J$
Replacement by "1"	$\mathbf{A}' = (\mathbf{a}, \mathbf{a}^c)$; *BUT*, set $a_i = a_i^c = 1$ if i missing		$\frac{\lvert \mathbf{w}_j \wedge \mathbf{A}' \rvert}{M} \geq \rho$	
Indicator vector	$\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$	$T_j(\mathbf{A}, \delta) = \frac{\lvert \mathbf{w}_j \wedge \mathbf{A} \wedge \delta \rvert}{\alpha + \lvert \mathbf{w}_j \wedge \delta \rvert}$	$\frac{\lvert \mathbf{w}_j \wedge \mathbf{A} \wedge \delta \rvert}{\lvert \mathbf{A} \wedge \delta \rvert} \geq \rho$	$\mathbf{w}'_J = \beta((\mathbf{A} \vee \delta^c) \wedge \mathbf{w}_J) + (1-\beta)\mathbf{w}_J$

Table 1: Modifications to fuzzy ARTMAP for implementation of missing component strategies.

random choice was made). Results are shown in Fig. 2. It can be seen that the indicator vector method performs better than replacement by "1" and much better than replacement by "0," whether components are missing during testing or training, while providing better compression than either.



(a) (b) (c)

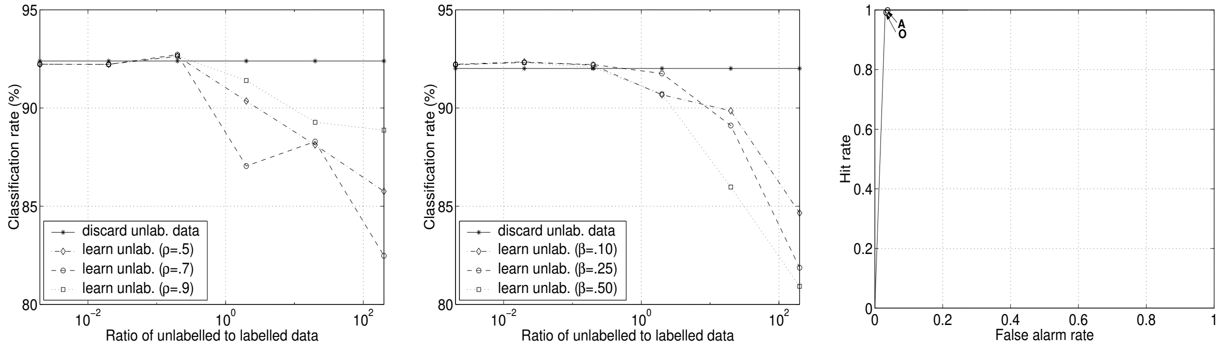
Figure 2: Missing input pattern components. (a) Accuracy with missing components during testing only. Fuzzy ARTMAP with indicator vector (top curve), replacement by "1" (middle curve), and replacement by "0" (bottom curve). (b) Same as (a), but missing components only during training. (c) Compression. Fuzzy ARTMAP with indicator vector (top curve), replacement by "1" (middle curve), and replacement by "0" (bottom curve).

5 Missing class labels during training

To examine the ability of fuzzy ARTMAP to handle training data with missing class labels, the network is trained in two phases. During the first phase, involving supervised learning, the network is trained with a fixed amount of labeled data (0.5% of the available training data). During the second phase, involving unsupervised learning, the network is presented with a variable amount of unlabeled data. Using the fuzzy ART algorithm (Carpenter *et al.*, 1991), with modifications described below, the network associates each unlabeled training input pattern with one of the already-existing internal categories and adjusts the weight vectors associated with that internal category as appropriate.

During the supervised-learning phase, the learning rate β and baseline vigilance $\bar{\rho}$ are kept at their respective default values $\beta = 1$ (fast learning) and $\bar{\rho}$=0. During the unsupervised-learning phase, smaller values of β and larger values of the fixed vigilance parameter ρ are used, as these have been found to improve the performance on the test set of the final trained classifier. Unlabeled patterns which cannot pass the vigilance test are discarded (*i.e.*, no new internal category nodes are allocated during the unsupervised-learning phase). Additional improvement in performance is obtained by applying, to any unlabeled pattern which has passed the vigilance test, a coactivation test. An unlabeled pattern is discarded if the activation level T_J of the node with which it is associated is not larger than the activation level $T_{J_{\rm next}}$ of the next-most-active node by a sufficiently great amount; *i.e.*, if $T_J - T_{J_{\rm next}} > \epsilon_{co}$ is not satisfied. (A value of $\epsilon_{co} = 0.05$ was used in the simulations.)

Fig. 3 shows the effect of retaining training data with missing class labels. Although the approach described above did, with suitable choice of parameters, substantially reduce degradation of the performance of the trained classifier due to the inclusion of unlabeled data, performance significantly *better* than that achieved by simply discarding all of the unlabeled training data was never observed.



(a) (b) (c)

Figure 3: (a,b) Missing class labels during training. $\beta = 0.1$ in (a) and $\rho = 0.7$ in (b) (see the text). (c) Missing classes. Typical ROC curve for familiarity discrimination without LUC. A is the actual operating point, O is the optimal operation point for the curve (minimum value of 1 - hit rate + false alarm rate).

6 Missing classes

The modification of fuzzy ARTMAP that deals with unfamiliar classes is ARTMAP-FD. This algorithm has been shown to effectively perform familiarity discrimination on simulated radar range profiles (Carpenter *et al.*, 1997) and radar emitter data (Granger *et al.*, 1999b).

For the simulations, 13 classes were selected out of the 15 emitter type classes, and labeled patterns from these 13 (familiar) classes were presented to the network during the learning phase. The operating threshold was determined during the learning phase using the online method (Carpenter *et al.*, 1997).

We first present the results of simulations in which no learning is allowed during the test phase. A hit rate of 99.7% and a false alarm rate of 3.2% were obtained. Accuracy on familiar-class patterns correctly flagged as such was 99.6%. The number of internal category nodes was 111. These results are averages over 20 selections of the 13 familiar classes. The selections were performed at random, with the restriction that selections leading to an insufficient number of unfamiliar-class test patterns (less than a thousand) were not allowed. A typical ROC curve from one of these selections is shown in Fig. 3(c).

We next present simulations in which learning continues during testing. The LUC algorithm employed in these simulations is as described in Section 1, with two modifications. To allow us to focus on the effects of LUC (as opposed to learning with missing class labels), the weights associated with internal category nodes allocated during the learning phase are kept at fixed values. In addition, patterns that are declared by the network to be from unfamiliar classes are given a "second chance" to be associated with an existing node before a new unlabeled node is allocated, in order to prevent the generation of an excessively large number of internal category nodes during testing.

Specifically, a pattern declared unfamiliar by the network is subjected to a vigilance test at each of the "new "nodes, *i.e.,* nodes that have been created during the test phase. If it passes this vigilance test, it is associated with the node with the highest vigilance value; *i.e.*, that node j out of all the new nodes for which $\mid \mathbf{A} \wedge \mathbf{w}_j \mid$ is largest. (In the simulations presented here, the vigilance parameter used was 0.8). No adjustment of the node's weights is performed.

If the pattern cannot in this way be associated with an already-existing new node, then a coactivation test is performed between the node J to which the pattern was tentatively assigned prior to having been declared unfamiliar and each of the new nodes j_{new}, using a small coactivation parameter $\epsilon_{co} = 0.05$. If $T_J - T_{j_{\text{new}}} < \epsilon_{co}$ for any j_{new}, the pattern is associated with the node j_{new} for which $T_J - T_{j_{\text{new}}}$ is smallest. (No weight adjustment takes place.) Only if neither of these options for association with an already-existing new node succeeds is a *new* new node created.

A hit rate of 99.8% and a false alarm rate of 3.3% were obtained with LUC. Accuracy on familiar-class patterns correctly flagged as such was 99.6%. The number of internal category nodes was 117. The Rand score for the new nodes was 0.783.

7 Conclusions and discussion

We conclude that, for the present application, fuzzy ARTMAP provides a high level of accuracy and compression even when the amount of training data is limited; that the indicator-vector method of dealing with missing components causes the least degradation in accuracy and compression when input component patterns are missing during training and/or testing; that the use of the vigilance and coactivation tests can prevent performance degradation when training on data with missing class labels (although improvement in performance was not seen on this data set); and that ARTMAP-FD familiarity discrimination can identify patterns belonging to unfamiliar classes during training and testing, and can allow learning of unfamiliar classes to take place during testing.

The importance for the application under consideration in this paper of being able to perform familiarity discrimination during the test (operational) phase is evident: radar emitters can exhibit new modes at any time. The ability to perform LUC during training is anticipated also to be of great importance for this application. Preliminary simulations indicate that LUC can improve performance in situations where "pure" FD performs poorly. Now, the task of providing training data for an emitter identification system involves slow, tedious labor by an ESM analyst, so it cannot be expected that more than a small fraction of the large amount of available data will be labeled for training. Furthermore, it cannot be assumed that all of the unlabeled training data comes from emitter classes that have been identified by the ESM analyst. With the modifications presented above, fuzzy ARTMAP should be able to mitigate performance degradation due to missing class labels, while being able to benefit by learning information hidden in the unlabeled data about as-yet unidentified classes.

Acknowledgments

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

Aguilar, J. M., and Ross, W. D., 1994, "Incremental ART: A Neural Network for Recognition by Incremental Feature Extraction," In *World Conference on Neural Networks—San Diego: 1994 International Neural Network Society Annual Meeting,* June 5-9 1994, Vol. I, 593-598.

Bensaid, A. M., Hall, L. O., Bezdek, J. C., and Clarke, L. P., 1996, "Partially Supervised Clustering for Image Segmentation," *Pattern Recognition*, **29**, 859-871.

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

Carpenter, G. A., Grossberg, S., and Rosen, D. B., 1991, "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System," *Neural Networks*, **4:6**, 759-771.

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

Carpenter, G. A, Rubin, M. A., and Streilein, W. W., 1997, "Threshold Determination for ARTMAP-FD Familiarity Discrimination." In C. H. Dagli *et al.,* eds., *Intelligent Engineering Systems Through Artificial Neural Networks,* Volume 7, pp. 23-28.

Demiriz, A., Bennett, K. P., and Embrechts, M. J., 1999, "Semi-Supervised Clustering Using Genetic Algorithms," In C. H. Dagli et al., eds., *Intelligent Engineering Systems Through Artificial Neural Networks 9*, New York, NY: ASME Press, 809-814.

Granger, E., Grossberg, S., Lavoie, P., and Rubin, M. A., 1999a, "Comparison of Classifiers for Radar Emitter Type Identification," In C. H. Dagli et al., eds., *Intelligent Engineering Systems Through Artificial Neural Networks 9*, New York, NY: ASME Press, 3-11.

Granger, E., Grossberg, S., Rubin, M. A., and Streilein, W. W., 1999b, "Familiarity Discrimination of Radar Pulses," In M. S. Kearns et al., eds., *Advances in Neural Information Processing Systems 11*, Cambridge, MA: MIT Press, 875-881.

Hubert, L., and Arabie, P., 1985, "Comparing Partitions," *Journal of Classification* **2**, 193-218.

Little, R. J. A., and Rubin, D. B., 1987, *Statistical Analysis with Missing Data.* New York: Wiley.

Wiley, R. G., 1993, *Electronic Intelligence: The Analysis of Radar Signals*, 2nd ed., Artech House.