Self-organizing Hierarchical Knowledge Discovery by an ARTMAP Image Fusion System

Gail A. Carpenter, Siegfried Martens, Ogi J. Ogas
Boston University
Department of Cognitive and Neural Systems
677 Beacon Street
Boston, MA 02215
USA
gail@cns.bu.edu, sig@cns.bu.edu, ogiogas@cns.bu.edu
http://cns.bu.edu/techlab

Abstract - *Classifying novel terrain or objects from sparse, complex data may require the resolution of conflicting information from sensors working at different times, locations, and scales, and from sources with different goals and situations. Information fusion methods can help resolve inconsistencies, as when evidence variously suggests that an object's class is* car, truck, *or* airplane. *The methods described here consider a complementary problem, supposing that information from sensors and experts is reliable though inconsistent, as when evidence suggests that an object's class is* car, vehicle, *and* man-made. *Underlying relationships among objects are assumed to be unknown to the automated system or the human user. The ARTMAP information fusion system uses distributed code representations that exploit the neural network's capacity for one-to-many learning in order to produce self-organizing expert systems that discover hierarchical knowledge structures. The system infers multi-level relationships among groups of output classes, without any supervised labeling of these relationships.*

Keywords: ARTMAP, Adaptive Resonance Theory (ART), information fusion, image fusion, data mining, remote sensing, distributed coding, association rules, multi-sensor fusion.

1 Introduction: Deriving consistent knowledge from inconsistent information

Image fusion has been defined as "the acquisition, processing and synergistic combination of information provided by various sensors or by the same sensor in many measuring contexts." [1, p. 3] When multiple sources provide inconsistent data, fusion methods are called upon to select the accurate information components. As quoted by the International Society of Information Fusion
(http://www.inforfusion.org/terminology.htm):
"Evaluating the reliability of different information sources is crucial when the received data reveal some inconsistencies and we have to choose among various options." For example, independent sources might label an identified vehicle *car* or *truck* or *airplane*. A fusion method could address this problem by weighing the confidence and reliability of each source, merging complementary information, or gathering more data. In any case, at most one of these answers is correct.

The methods described here address a complementary and previously unexamined aspect of the information fusion problem, seeking to derive consistent knowledge from sources that are inconsistent – but accurate. This is a problem that the human brain solves very well. A young child who hears the family pet variously called *Spot*, *puppy*, *dog*, *dalmatian*, *mammal*, and *animal* is not only not alarmed by these labels but readily uses them to infer functional relationships. An analogous problem for information fusion methods seeks to classify the terrain and objects in an unfamiliar territory based on intelligence supplied by several reliable experts. Each expert labels a portion of the region based on sensor data and observations collected at specific times and based on individual goals and interests. Across experts, a given pixel might be correctly but inconsistently labeled *car*, *vehicle*, and *man-made*. A human mapping analyst would, in this case, be able to apply a lifetime of experience to resolve the paradox by placing objects in a knowledge hierarchy, and a rule-based expert system could be constructed to codify this knowledge.

The current study shows how an ARTMAP neural network can act as a self-organizing expert system to derive hierarchical knowledge structures from inconsistent training data. This ability is implicit in the network's learning strategy, which creates one-to-many, as well as many-to-one, maps of the input space. During training, the system can learn that disparate pixels map to the output class *car*; but, if similar or identical pixels are later labeled *vehicle* or *man-made*, the system can associate multiple output classes with a given input. During testing, distributed code activations predict multiple output class labels. A rule-production algorithm uses these distributed outputs to derive a knowledge hierarchy for the output classes. The resulting diagram of the relationships among classes can then guide the construction of consistent layered maps.

Sec. 2 outlines how distributed coding in the default ARTMAP network supports many-to-one and one-to-many learning. Sec. 3 describes a remote sensing testbed example, with sensor data from the Boston area. Sec. 4 specifies the algorithm that derives hierarchical knowledge structures from the trained network's distributed output class predictions, and Sec. 5 demonstrates system performance on multiband sensor data derived from the Boston area. Sec. 6 points to the application of ARTMAP fusion methods in other application domains.

2 Multi-class predictions by ARTMAP networks

Adaptive Resonance Theory (ART) neural networks model real-time prediction, search, learning, and recognition. ART networks function both as models of human cognitive information processing (e.g., [2–7]) and as neural systems for technology transfer (e.g., [8–10]). Sites of early and ongoing transfer of ART-based technologies include industrial venues such as the Boeing Corporation [11]. A recent report on industrial uses of neural networks [12] states: "[The] Boeing … Neural Information Retrieval System is probably still the largest-scale manufacturing application of neural networks. It uses [ART] to cluster binary templates of aeroplane parts in a complex hierarchical network that covers over 100,000 items, grouped into thousands of self-organised clusters. Claimed savings in manufacturing costs are in millions of dollars per annum."

Design principles derived from scientific analyses and design constraints imposed by targeted applications have jointly guided the development of many variants of the basic networks, including fuzzy ARTMAP [13], simplified fuzzy ARTMAP [14], ART-EMAP [15], ARTMAP-IC [16], Gaussian ARTMAP [17], and distributed ARTMAP [18]. Across many variations of these models, a neural computation central to both the scientific and the technological analyses is the *ART matching rule* [19], which represents the interaction between top-down learned expectation and bottom-up sensory input. This interaction creates a focus of attention which, in turn, determines the nature of stored memories.

While the earliest unsupervised ART [19] and supervised ARTMAP networks [20] feature winner-take-all code representations, many of the networks developed over the past ten years incorporate distributed code representations. Comparative analyses of these systems have led to the specification of a *default ARTMAP* network, which features simplicity of design and robust performance in many application domains [21]. Selection of one particular *a priori* algorithm is intended to facilitate technology transfer. This network, which here serves as the recognition engine of the information fusion system, uses winner-take-all coding during training and distributed coding during testing. Distributed test outputs have helped improve various methods for categorical decision-making. One such method, in a map production application, compares a baseline mapping procedure, which selects the class with the largest total output, with a procedure that enforces *a priori* output class probabilities and one that selects class-specific output thresholds, via validation [22]. Distributed coding supports each method, but the ultimate prediction is still one output class per test input. This paper also specifies a canonical training / testing method, which partitions the area in question into four vertical or horizontal strips. A given simulation takes training pixels from two of these strips; uses the validation strip to choose parameters, if necessary; and tests on the fourth strip. Methods are thus compared with training and test sets that are not only disjoint but drawn from geographically distinct locations. This separation tests for generalization to new regions, where class distributions could typically be far from those of the training and validation sets.

Fig. 1. Boston image testbed representation of preprocessed image: The city of Revere is at the center, surrounded by (clockwise from lower right) portions of Winthrop, East Boston, Chelsea, Everett, Malden, Melrose, Saugus, and Lynn. Logan Airport runways and Boston Harbor are at the lower center, with Revere Beach and the Atlantic Ocean at the right. The Saugus and Pines Rivers meet in the upper right, and the Chelsea River is in the lower left of the image. Landsat 7 spectral band values were acquired from the Earth Resources Observation System (EROS) Data Center, U.S. Geological Survey, Sioux Falls, SD (http://edc.usgs.gov/). Dimensions: 180 x 300 pixels (30*m*/pixel resolution) ≅ 5.4 *km* x 9 *km*.

The information fusion techniques summarized below modify the baseline mapping procedure by allowing the system to predict more than one output class during testing. A given pixel either predicts the N classes receiving the largest net system outputs or predicts all classes whose net output exceeds a designated threshold Γ. For either multi-class prediction method, the parameter N or Γ is chosen from the validation strip.

3 Boston testbed example

The Boston testbed (Fig. 1) was derived from a Landsat 7 Thematic Mapper (TM) image acquired on the morning of January 1, 2001. The 5.4*km* x 9*km* area includes portions of northeast Boston and suburbs. The resolution of the Boston image is $30m^2$ in six TM bands, $60m^2$ in two thermal bands, and $15m^2$ in one Panchromatic band. The Boston image region encompasses mixed urban, suburban, industrial, water, and park spaces. Ground truth pixels were labeled: *ocean*, *ice*, *river*, *beach*, *park*, *road*, *residential*, *industrial*, *water*, *open space*, *built-up*, *natural*, *man-made*.

Inputs for the Boston example were preprocessed by a version of the Lincoln Lab image mining system [23-25], called the *Neural Fusion Module*, which was developed by Waxman and colleagues working in the CNS Technology Laboratory during 2001-2002 [26, 27]. For each pixel in the Boston image, this Module, implemented on an ERDAS Imagine (http://gis.leica-geosystems.com) platform, produced a 41-dimensional input vector representing local contrast, color, and texture attributes.

4 Deriving a knowledge hierarchy from a trained network: Predictions, rules, and graphs

The *ARTMAP fusion system* provides a canonical procedure for labeling an arbitrary number of output classes in a supervised learning problem. A critical aspect of the embedded default ARTMAP network is the distributed nature of its internal code representation, which produces continuous-valued test set predictions distributed across output classes (Sec. 2). Following a canonical partitioning procedure, each image was divided into four vertical strips. In the Boston example, training pixels were drawn from strips 1 and 3, validation pixels from strip 4, and test pixels from strip 2. Note, for example, the different distributions of the *water* class across vertical strips in the image (Fig. 1). Each training set contained a fixed number of pixels for each output class; or, for rare classes such as *road*, the training set contained all pixels available in the labeled ground truth subset of the training strip.

Information implicit in the distributed predictions of a trained ARTMAP network can be used to generate a hierarchy of output class relationships. To accomplish this, each test set pixel first produces a set of output class predictions (Sec. 4.1). The resulting list of test predictions then determines a list of rules $x \Rightarrow y$, which define relationships between pairs of output classes, with each rule carrying a confidence value (Sec. 4.2). The rules are then used to assign classes to *levels*, with rule antecedents x at lower levels and consequents y at higher levels (Sec. 4.3). Classes connected by arrows that codify the list of rules and confidence values form a graphical representation of the knowledge hierarchy.

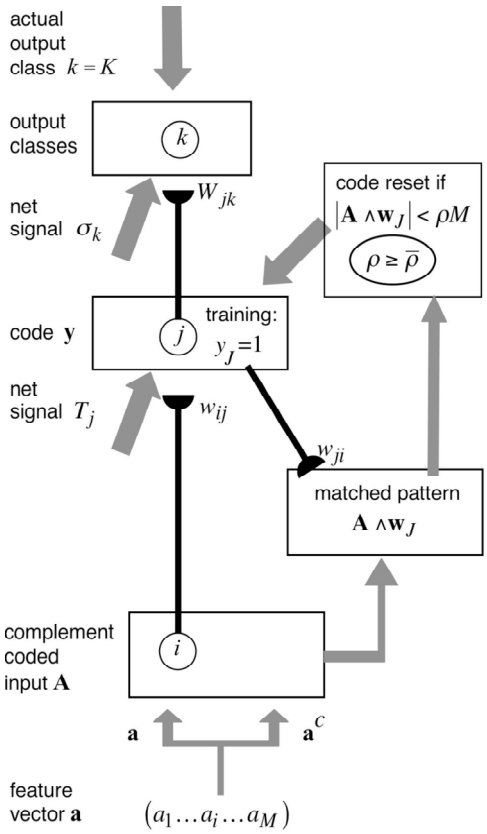

Fig. 2. Default ARTMAP notation: An M-D feature vector $\mathbf{a}$ is complement coded to form the $2M$-D ARTMAP input $\mathbf{A}$. Vector $\mathbf{y}$ represents a winner-take-all code during training, when a single category node (j=J) is active; and a distributed code during testing. With fast learning, bottom-up weights w_{ij} equal top-down weights w_{ji}, both represented by weight vector $\mathbf{w}_j$. Each coding node j is connected to a single output class node k, for which W_{jk}=1. A distributed code $\mathbf{y}$ thereby produces predictions σ_k distributed across output classes. In all simulations reported here, the baseline vigilance matching parameter is set to its default value, $\bar{\rho} = 0$. [21]

4.1 Predictions

In response to a test input, distributed activations in the default ARTMAP coding field send a net signal σ_k to each output class k (Fig. 2). Competitive normalization of the code **y** implies that the total system signal to all output classes is also normalized:

$$\sum_k \sigma_k = \sum_k [\sum_{j:W_{jk}=1} y_j] = \sum_j y_j = 1$$

A baseline method predicts the single output class $k=K$ receiving the largest signal σ_k. Alternatively, a single test input can predict multiple output classes, according to two methods tested here. A *threshold method* predicts all output classes k for which σ_k exceeds a signal threshold Γ. A *TopN method* forces each pixel to choose the N classes with the largest signals σ_k.

The optimal value of the *prediction parameter* Γ or N is estimated from a subset of pixels drawn from the images's validation strip. Recall that the ground truth set may assign any number of output class labels to a given pixel, but that the system has no knowledge of multi-class relationships during its incremental learning phase. During testing, each input pixel tends to make more predictions as the threshold Γ for the distributed output pattern decreases (or as the number of TopN predictions increases). Typically, a high threshold Γ yields few predictions per pixel, but these few predictions are likely to be "correct," or *hits*, i.e., they are among those specified by the ground truth set. A high prediction threshold thus implies low *recall*, defined as the average across pixels of the number of hits divided by the total number of labels specified by the ground truth set; but high *precision*, defined as the average across pixels of the number of hits divided by the number of labels predicted by the network. Conversely, a low threshold Γ tends to discover most of the output classes in the ground truth set, producing a high recall rate; but at a cost of predicting many additional, incorrect classes, producing low precision.

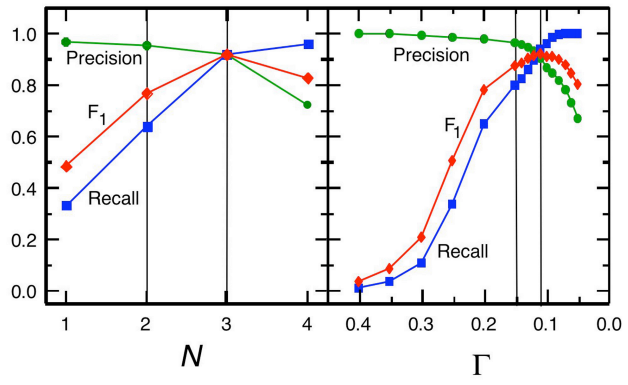

Fig. 3. On validation subsets of the Boston testbed, increasing the number N of predicted output classes per pixel, or decreasing the output threshold Γ, produces higher rates of recall (■) but lower precision (●). The F_1 measure (◆) trades-off the competing goals of high precision and high recall, to produce estimates of the optimal prediction parameter values. Vertical lines indicate values of N and Γ for simulations in Sec. 5.

Values of Γ and N are chosen so as to generate as many of the ground truth labels as possible (high recall) without sacrificing accuracy (high precision). A common method for balancing these two goals maximizes the F_1 *measure* [28, 29], which is defined as:

$$F_1 = 2\frac{precision \times recall}{precision + recall}$$

Note that F_1 is symmetric with respect to recall and precision, and lies between these two quantities; and that at the *cross-over point* where precision equals recall, F_1 equals their common value.

Fig. 3 illustrates how recall, precision, and F_1 values vary on validation subsets of pixels drawn from the Boston image with increasing N, for the *TopN* prediction method, and with decreasing output class thresholds Γ. Graphs of recall and precision derived from a validation subset of the Boston testbed point to optimal values of N=3 and $\Gamma \cong 0.11$, each being close to the crossover point where recall equals precision and to the peak of the graph of F_1.

4.2 Rules

Based on the validation set analysis of recall and precision (Sec. 4.1), a user determines a prediction parameter equal to a fixed number N of output classes per pixel or an output signal threshold Γ. Each test pixel produces a set of output class predictions $\{x, y, \ldots\}$ from its distributed signals σ_k, according to the chosen method. The list of all multi-class test set predictions is then used to deduce a list of output class implications of the form $x \Rightarrow y$, each carrying a confidence value C%. This rule-creation method is related to the Apriori algorithm of Agrawal and Srikant [30, 31] for generating data mining association rules.

The following steps derive the list of rules. The algorithm introduces an *equivalence parameter e%* and a *minimum confidence parameter c%*. Two classes x and y are treated as *equivalent* $(x \equiv y)$ if both rules $x \Rightarrow y$ and $y \Rightarrow x$ hold with confidence greater than e. In this case, the class predicted by fewer pixels is ignored in subsequent computations, but equivalent classes are displayed as a single node on the final rule summary graph. Rules with low confidence ($C < c$) are ignored, with one exception: if all rules that include a given class have confidence below c, then the list retains the rule derived from the pair predicted by the largest number of pixels. Although this "no extinction" clause may produce low-confidence rules, these tend to correspond to cases that are rare but important. The user can easily take these exceptions under advisement, since the graph displays each confidence value.

Reasonable default values set the equivalence parameter e between 90-95%, and the minimum confidence parameter c between 50-70%. In all simulations shown here e=91% and c=50%. Alternatively, e and c may chosen by validation.

Rule Step 1 List the number of test set pixels predicting each output class x. Order this list from the classes with the fewest predictions to the classes with the most.

Rule Step 2 List the number of test set pixels $\#(x \,\&\, y)$ simultaneously predicting each pair of distinct output classes. Omit pairs with no such pixels. Order the list so that $\#(x) \le \#(y)$: classes x observe the order established in Rule Step 1; and for each such class x, classes y observe the same order.

Rule Step 3 Identify equivalent classes, where $x \equiv y$ if $[\#(x \,\&\, y) / \#(y)] \ge e\%$. Remove from the list all class pairs that include x.

Rule Step 4 Each pair remaining on the list produces a rule $x \Rightarrow y$ with confidence $C\% = [\#(x \,\&\, y) / \#(x)]$. If Rule Step 3 determined that $x \equiv y$, record the confidence $C \ge e$ of each rule in the pair $\{x \Rightarrow y,\ y \Rightarrow x\}$.

Rule Step 5 Remove from the list all rules with confidence $C < c$. *Exception (no extinction)*: If all rules that include a given class have confidence below the minimum confidence c, then retain the rule or rules $x \Rightarrow y$ with maximal $\#(x \,\&\, y)$ pixels.

Rule Step 6 The following optional information may be useful for purposes of analysis.

(a) List rules removed in Rule Step 5 that have confidence in a marginal range, say $10\% \le C < c$.

(b) List class pairs $x \,\&\, y$ (from Rule Step 2) with equivalence values in a marginal range. For example, list the rule pairs $\{x \Rightarrow y,\ y \Rightarrow x\}$ for class pairs $x \,\&\, y$ for which $c \le [\#(x \,\&\, y) / \#(y)] < e$.

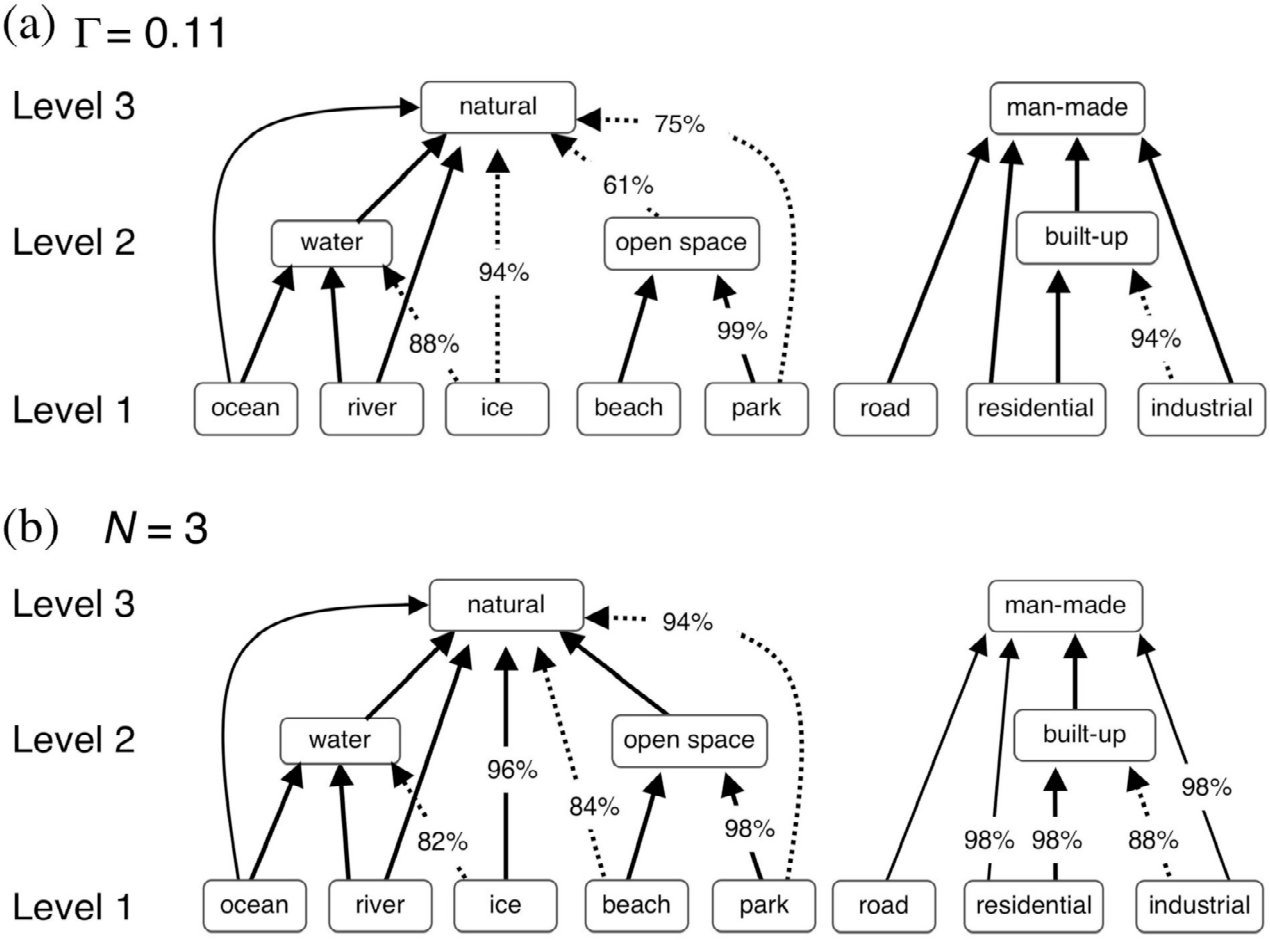

Fig. 4. Boston testbed knowledge hierarchies derived by the ARTMAP fusion system for prediction parameters corresponding to peak F_1 validation set values: (a) The threshold $\Gamma = 0.11$ produces all the correct rules except *beach* $\Rightarrow$ *natural* (C=19%); plus eight other marginal rules and five marginal equivalence relations, all incorrect. (b) Setting N=3 produces all the correct rules; plus five marginal rules and three marginal equivalence relations, all incorrect.

4.3 Graphs

A directed graph summarizes the list of rules derived in Sec. 4.2. These rules suggest a natural hierarchy among output classes, with antecedents sitting below consequents. For each rule $x \Rightarrow y$, class x is located at a lower level of the hierarchy than class y, according to the iterative algorithm below. Once each class is situated on its level, a listed rule $x \Rightarrow y$ produces an arrow from x to y. Each rule's confidence is indicated by the arrow, with lower-confidence rules (say C<95%) portrayed by dashed arrows. For arrows with no displayed confidence values, C=100%.

The following steps assign each output class to a level.

Top Level Items that appear only as consequents y.

Level 1 Classes that do not appear as consequents in any rule.
Remove from the list all rules $x \Rightarrow y$ where x is in Level 1.

Next Level Classes that do not appear as consequents in any remaining rule.
Remove from the list all rules $x \Rightarrow y$ where x is in this level.

Iterate Repeat until all rules have been removed from the list.

Note that Level 1 includes classes that do not appear in any rule as well as those that appear only as antecedents.

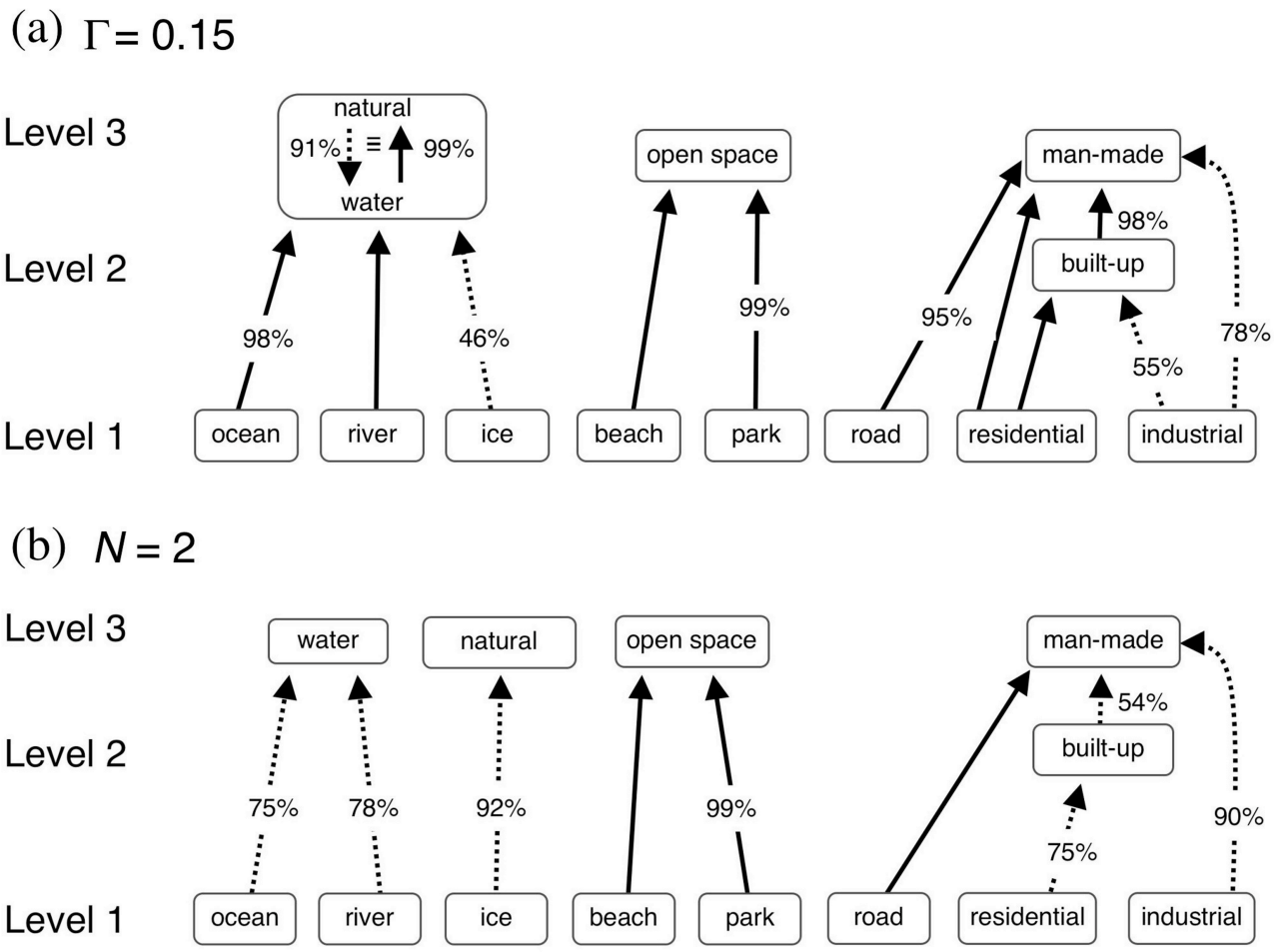

Fig. 5. Boston graphs for prediction parameter values with high precision but low recall: (a) The too-high threshold Γ=0.15 produces the incorrect equivalence relation *water* ≡ *natural*. A slightly higher value of the equivalence parameter e would have restored the rule *water* ⇒ *natural*, and produced a graph that more closely reflects the true hierarchy. Note that the no-extinction clause in Rule Step 5 preserves the correct (but low confidence) rules *ice* ⇒ *water* and *ice* ⇒ *natural*. (b) The too-small output class number N = 2 misses a number of rules, but still discovers at least part of the true three-level hierarchy.

5 Graphical representations of knowledge hierarchies

Graphs in Fig. 4 depict the rules derived for the Boston example with prediction parameters set to the optimal levels chosen by validation. Graphs in Fig. 5 show how performance deteriorates when parameters are set above or below their optimal values.

5.1 Optimal prediction parameters for the Boston testbed

Setting Γ = 0.11 maximizes the F_1 measure for the Boston image (Fig. 3). Fig. 4a shows that this threshold value places each class in its correct level, and discovers all the correct rules except for *beach* $\Rightarrow$ *natural*.

For the TopN method, setting $N = 3$ (Fig. 4b) produces all the correct rules in a complete graph. Note that in the Boston ground truth set, *road* pixels are also labeled *man-made*, while all other ground truth pixels have three class labels.

5.2 Sub-optimal prediction parameters for the Boston testbed

Fig. 5a shows that, with the high threshold Γ=0.15, the ARTMAP fusion system misses the rule *open space* $\Rightarrow$ *natural* in the Boston example. It therefore equates *natural* with *water*. Note that the (correct) rule pair $\{$ *ice* $\Rightarrow$ *natural*, *ice* $\Rightarrow$ *water* $\}$ is included in the graph despite its low confidence (C = 46%). This is due to the no-extinction clause in Rule Step 5 (Sec. 4.2), which ensures that the rule with maximal confidence involving each class survives pruning. Despite the high output threshold, the system is able to infer the transitive relationships *residential* $\Rightarrow$ *built-up* $\Rightarrow$ *man-made* and *industrial* $\Rightarrow$ *built-up* $\Rightarrow$ *man-made*.

Compared to the threshold method with Γ=0.15, setting $N = 2$ produces even fewer rules, and a somewhat different graph (Fig. 5b). All specified rules are again correct. The $N = 2$ system also correctly derives the transitive relationship *residential* $\Rightarrow$ *built-up* $\Rightarrow$ *man-made*.

6 Conclusion: ARTMAP information fusion

The ARTMAP neural network produces one-to-many mappings from input vectors to output classes, as well as many-to-one mappings, as the normal product of its supervised learning laws. During training, a given input may be associated with more than one output class. Some of these associations could be erroneous: when different observers label an image *dog*, *coyote*, or *wolf*, at most one of these classes is correct. Inconsistent data may, however, be completely correct, as when observers variously label the image *wolf*, *mammal*, and *carnivore*. By resolving such paradoxes during everyday knowledge acquisition, humans naturally infer complex, hierarchical relationships among classes without explicit supervision. One-to-many learning allows the ARTMAP information fusion system to associate any number of output classes with each input. Although inter-class relationships are not specified with the training inputs, the system readily derives knowledge of the rules, confidence estimates, and multi-class hierarchical relationships from patterns of distributed test predictions.

The testbed example from the Boston image demonstrates how ARTMAP information fusion resolves apparent contradictions in input pixel labels by assigning output classes to levels in a knowledge hierarchy. This methodology is not limited to the image domain illustrated here, and could be applied, for example, to infer patterns of drug resistance from medical data or to improve marketing suggestions to individual consumers.

Acknowledgements

This work was supported by research grants from the Air Force Office of Scientific Research (AFOSR F49620-01-1-0397 and AFOSR F49620-01-1-0423) and the Office of Naval Research (ONR N00014-01-1-0624); by postdoctoral fellowships from the National Imagery and Mapping Agency and the National Science Foundation for Siegfried Martens (NMA501-03-1-2030 and NSF DGE-0221680); and by a Department of Homeland Security graduate fellowship for Ogi Ogas. Sucharita Gopal, Junchang Ju, and Mutlu Ozdogan provided remote sensing data, and Suhas Chelian and Brad Rhodes helped create the ground truth sets for the Boston testbed.

References

[1] G. Simone, A. Farina, F.C. Morabito, S.B. Serpico, and L. Bruzzone. Image fusion techniques for remote sensing applications. *Information Fusion*, 3:3-15, 2002.

[2] G.A. Carpenter. Distributed learning, recognition, and prediction by ART and ARTMAP neural networks. *Neural Networks*, 10(8):1473-1494, 1997.

[3] G.A. Carpenter and S. Grossberg. Normal and amnesic learning, recognition, and memory by a neural model of cortico-hippocampal interactions. *Trends in Neuroscience*, 16(4):131-137, 1993.

[4] S. Grossberg. How does a brain build a cognitive code? *Psychological Review*, 87:1-51, 1980.

[5] S. Grossberg. The link between brain, learning, attention, and consciousness. *Consciousness and Cognition,* 8:1-44, 1999.

[6] S. Grossberg. How does the cerebral cortex work? Development, learning, attention, and 3-D vision by laminar circuits of visual cortex. *Behavioral and Cognitive Neuroscience Reviews*, 2(1):47-76, 2003.

[7] M. Page. Connectionist modelling in psychology: A localist manifesto. *Behavioral and Brain Sciences*, 23:443-512, 2000.

[8] R.K. Aggarwal, Q.Y. Xuan, A.T. Johns, F. Li, and A. Bennett. A novel approach to fault diagnosis in multicircuit transmission lines using fuzzy ARTMAP neural networks. *IEEE Trans. on Neural Networks*, 10(5):1214-1221, 1999.

[9] S. Gopal, C. Woodcock, and A. Strahler. Fuzzy ARTMAP classification of global land cover from the 1 degree AVHRR data set. *Remote Sensing of Environment*, 67:230-243, 1999.

[10] N. Griffith and P.M. Todd, editors. Musical Networks: Parallel Distributed Perception and Performance. MIT Press, Cambridge, MA, 1999.

[11] T.P. Caudell, S.D.G. Smith, R. Escobedo, and M. Anderson. NIRS: Large scale ART 1 neural architectures for engineering design retrieval. *Neural Networks*, **7**(9):1339-1350, 1994.

[12] P. Lisboa. Industrial use of safety-related artificial neural networks. *Contract Research Report 327/2001*, John Moores University, Liverpool, UK, 2001.

[13] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3(5):698-713, 1992.

[14] T. Kasuba. Simplified fuzzy ARTmap. *AI Expert*, 8(11):18-25, 1993.

[15] G.A. Carpenter and W.D. Ross. ART-EMAP: a neural network architecture for object recognition by evidence accumulation. *IEEE Transactions on Neural Networks*, 6(4):805-818, 1995.

[16] G.A. Carpenter and N. Markuzon. ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases. *Neural Networks*, 11(2):323-336, 1998.

[17] J.R. Williamson. Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, 9(5):881-897, 1998.

[18] G.A. Carpenter, B.L. Milenova, and B.W. Noeske. Distributed ARTMAP: a neural network for fast distributed supervised learning. *Neural Networks,* 11(5):793-813, 1998.

[19] G.A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54-115, 1987.

[20] G.A. Carpenter, S. Grossberg, and J.H. Reynolds, J.H. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4(5):565-588, 1991.

[21] G.A. Carpenter. Default ARTMAP. In *Proc. Int. Joint Conf. on Neural Networks (IJCNN'03)*, pages 1396-1401, Portland, Oregon, 2003.

[22] O. Parsons and G.A. Carpenter. ARTMAP neural networks for information fusion and data mining: map production and target recognition methodologies. *Neural Networks*, 16(7):1075-1089, 2003.

[23] W.D. Ross, A.M. Waxman, W.W. Streilein, M. Aguilar, J. Verly, F. Liu, M.I. Braun, P. Harmon, and S. Rak. Multi-sensor 3D image fusion and interactive search. In *Proc. Third Int. Conf. Information Fusion*, Paris, vol. I, 2000.

[24] W. Streilein, A. Waxman, W. Ross, F. Liu, M. Braun, D. Fay, P. Harmon, and C.H. Read. Fused multi-sensor image mining for feature foundation data. In *Proc. Third Int. Conf. Information Fusion*, Paris, vol. I, 2000.

[25] A.M. Waxman, J.G. Verly, D.A. Fay, F. Liu, M.I. Braun, B. Pugliese, W.D. Ross, and W.W. Streilein. A prototype system for 3D color fusion and mining of multisensor / spectral imagery. In *Proc. Fourth Int. Conf. Information Fusion*, pages 2-10, Montreal, vol. I, 2001.

[26] A.M. Waxman, D.A. Fay, B.J. Rhodes, T.S. McKenna, R.T. Ivey, N.A. Bomberger, V.K. Bykoski, and G.A. Carpenter. Information fusion for image analysis: geospatial foundations for higher-level fusion. In *Proc. Fifth Int. Conf. Information Fusion*, Annapolis, 2002.

[27] D.A. Fay, R.T. Ivey, N. Bomberger, and A.M. Waxman. Image fusion and mining tools for a COTS environment. In *Proc. Sixth Int. Conf. Information Fusion*, pages 606-613, Cairns, Australia, 2003.

[28] M.E. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1):87-118, 2002.

[29] C.J. van Rijsbergen. *Information Retrieval, Second Edition*. Butterworth-Heinemann, Newton, MA, 1981.

[30] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD Conf.*, Washington, DC, 1993.

[31] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. Twentieth Int. Conf. on Very Large Data Bases (VLDB)*, Santiago, Chile, 1994.