Unifying multiple knowledge domains using the ARTMAP information fusion system

Gail A. Carpenter, Arun Ravindran

Department of Cognitive and Neural Systems
Boston University
677 Beacon Street
Boston, Massachusetts 02215 USA

[gail, aravindr] @cns.bu.edu

Submitted to: *Proceedings of the 11th International Conference on Information Fusion*
Cologne, Germany
June 30 – July 3, 2008

Technical Report CAS/CNS TR-2008-001
Boston, MA: Boston University

Acknowledgements : This work was supported by research grants from the National Geospatial-Intelligence Agency (NMA 201-01-1-2016) and the National Science Foundation (NSF SBE-0354378).

Unifying multiple knowledge domains using the ARTMAP information fusion system

Gail A. Carpenter
Department of Cognitive and Neural Systems
Boston University
Boston, MA 02215, USA
gail@cns.bu.edu

Arun Ravindran
Department of Cognitive and Neural Systems
Boston University
Boston, MA 02215, USA
aravindr@cns.bu.edu

Abstract - Sensors working at different times, locations, and scales, and experts with different goals, languages, and situations, may produce apparently inconsistent image labels that are reconciled by their implicit underlying relationships. Even when such relationships are unknown to the user, an ARTMAP information fusion system discovers a hierarchical knowledge structure for a labeled dataset. The present paper addresses the problem of integrating two or more independent knowledge hierarchies based on the same low-level classes. The new system fuses independent domains into a unified knowledge structure, discovering cross-domain rules in this process. The system infers multi-level relationships among groups of output classes, without any supervised labeling of these relationships. In order to self-organize its expert system, ARTMAP information fusion system features distributed code representations that exploit the neural network's capacity for one-to-many learning. The fusion system software and testbed datasets are available from http://cns.bu.edu/techlab

Keywords: ARTMAP, Adaptive Resonance Theory (ART), information fusion, image fusion, data mining, remote sensing, distributed coding, expert system, neural network.

1 Introduction: Deriving consistent information from inconsistent sources and experts

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 and experts provide inconsistent data, fusion methods are called upon to select the accurate information components. 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 derive consistent knowledge from sources that are inconsistent but accurate and from experts who have different goals consistent with their domain knowledge. This is a problem that the human brain solves very well. A young child who hears the family pet variously called *Sarah's dog, Fluffy, Samoyed, dog*, *mammal*, and *animal* is not only not alarmed by these labels but readily uses them to infer functional relationships (Figure 1). Another child might learn the partially overlapping labels *chien, dog, pet,* and *animal* – as well as the incorrect label *wolf*.

The ARTMAP information fusion system acts as a self-organizing expert system to derive hierarchical knowledge structures from inconsistent training data [2, 3]. 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 on an image, for example, the system can learn that disparate pixels map to the output class *ocean*; but, if similar or identical pixels are later labeled *water* or *natural*, the system learns to associate multiple output classes with a given input. During testing, distributed code activations predict multiple output class labels. A rule-discovery algorithm uses these distributed outputs to derive a knowledge hierarchy for the output classes (Figure 1). The resulting diagram of the relationships among classes can then guide the construction of consistently layered maps.

The current study addresses the problem of fusing multiple knowledge domains. For example, Figure 1 illustrates a domain in which one expert labels portions of an image as *man-made* versus *natural.* Another expert working in another domain might label the same region as *wet* versus *dry*. Yet another expert might label the same region with more labels, e.g., *ocean, river, ..., industrial.* The system described in this paper fuses information implicit in the *natural/man-made* example in Figure 1 with information from another domain defined by *wet/dry* at the top-level. The unified system discovers cross-domain rules such as *wet*$\Rightarrow$*natural* and *man-made*$\Rightarrow$*dry.* In addition, the cross-domain system discovers equivalence relationships, as when an area is labeled *wet* in one domain and *water* in another.

Section 2 outlines how distributed coding in the default ARTMAP network supports many-to-one and one-to-many learning. Section 3 describes two remote sensing testbed examples. Section 4 specifies the algorithm that derives hierarchical knowledge structures from the trained

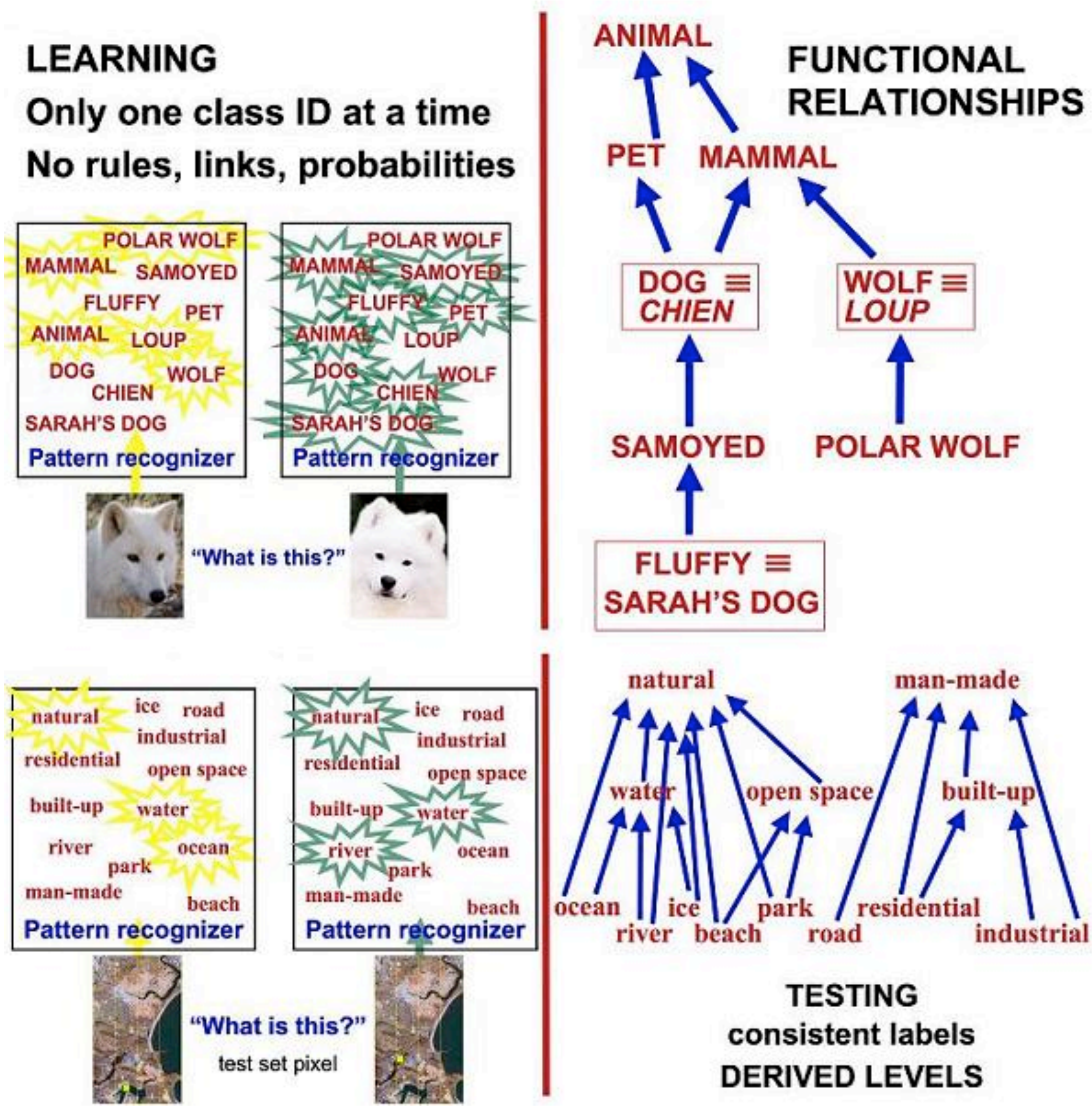

Figure 1 In the ARTMAP information fusion system, each training set pixel is associated with only one label, but test set pixels make multiple predictions. The system derives a hierarchical knowledge structure from these distributed predictions.

network's distributed output class predictions, and Section 5 demonstrates system performance on multiband sensor data. Section 6 shows how an extension of the single domain ARTMAP information fusion system unifies multiple knowledge domains.

2 Multi-class predictions by ARTMAP neural 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., [4–8]) and as neural systems for technology transfer (e.g., [9–13]). 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 [14], simplified fuzzy ARTMAP [15], Gaussian ARTMAP [16], and distributed ARTMAP [17]. Across many variations of these models, a neural computation central to both the scientific and the technological analyses is the ART matching rule [18], 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 [18] and supervised ARTMAP networks [19] feature winner-take-all code representations, many of the networks developed over the past fifteen 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 [20, 21].

Selection of one particular *a priori* algorithm is intended to facilitate technology transfer. The default ARTMAP network, which 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.

3 Boston and Monterey testbed examples

The methods developed here are illustrated with two image testbed examples.

The Boston testbed [22], derived from a Landsat 7 Thematic Mapper (TM), is an image of a 5.4*km* x 9*km* area that 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 are labeled: *ocean*, *ice*, *river*, *beach*, *park*, *road*, *residential*, *industrial*. The Boston testbed is available from http://cns.bu.edu/techlab.

The Monterey testbed is based on an aerial photograph of the Monterey Naval Postgraduate School at 0.5 m resolution. Ground truth pixels are labeled: *cars, roof, road, foot path, grass, trees*.

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. Information implicit in the distributed predictions of a trained ARTMAP network generates a hierarchy of output class relationships. To accomplish this, each test set pixel first produces a set of output class predictions (Section 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 (Section 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 (Section 4.3). Classes connected by arrows that codify the list of rules and confidence values form a graphical representation of the knowledge hierarchy.

4.1 ARTMAP fusion system training protocol

This section describes the cross-validation, training set selection, post-processing, and voting procedures employed by the ARTMAP information fusion system.

According to a standardized cross-validation procedure [22], the Boston image is divided into four vertical strips. Training pixels are drawn from three of the strips and test pixels from another strip. A single system would be trained, for instance, on pixels from Strips 1, 3, and 4 and tested on pixels from Strip 2. Note the challenge presented by the different distributions of classes, such as *water*, across vertical strips in the Boston image. The simulations reported here are the result of cross-validation across the four possible train/test strip combinations.

A ground truth set typically includes some classes, such as *ocean,* with many pixels and some classes, such as *road*, with few pixels. In order for the learning system to encode underrepresented exemplars, the training protocol imposes a cap (set to 150 for the Boston testbed and 250 for the Monterey testbed) on the maximum number of labels from each class. During training, once a class reaches its cap no more pixels are associated with this class.

When an ARTMAP training input activates a coding node j for the first time, this node is said to become *committed*, and the output weight W_{jk} from node j to the associated output class k is set equal to 1 for the duration of training. This procedure partitions the coding nodes according to the output class to which they were first linked. A post-processing training step [17] presents the input-output pairs once more, this time distributing the activations $\mathbf{y}$ at the coding field and hence also distributing the output predictions. The output weights W_{jk} are then retrained to minimize the total least-squared error between predicted and actual outputs. This procedure is akin to the second stage of training in a radial basis function network. Final weights W_{jk} are here computed in batch mode using the Matlab pseudo-inverse function pinv.

Finally, many applications benefit from voting across several ARTMAP systems produced by a given training set. This feature derives from the fact that fast learning produces different networks, and hence different error patterns, for different input orderings. Simple voting procedures typically produce improved accuracy compared to that of any single network, with five voters normally serving as a good default choice.

4.2 Predictions

A critical aspect of the default ARTMAP network is the distributed nature of its internal code representation, which produces continuous-valued predictions across output classes during testing. 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 (Figure 2). A winner-take-all method predicts the single output class k=K receiving the largest signal σ_k. Alternatively, a single test input can predict multiple output classes. The per-pixel filtering method [3] allows the output activation pattern produced by each test pixel to determine the number of predicted classes. Namely, if the net signals σ_k projecting to the output classes k are arranged from largest to smallest, the system predicts all the classes up to the point of maximum decrease in the signal size from one class to the next. This strategy is motivated by the behavior of a hypothetical system that accurately represents all the output classes. In such a system, if a pixel should predict three classes (e.g., *river, water, natural*), then the output signals σ_k to each of these classes would typically be large compared to those of the remaining classes. The maximum decrease in size would then occur between the third and fourth largest signals, and the per-pixel filtering method would predict three classes.

4.3 Rules

Once each test pixel has produced a set of output class predictions $\{x,y,\ldots\}$, according to the per-pixel selection method, the list of multi-valued 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 in the association rule literature [23]. The steps listed below produce the list of rules that label class relationships. The algorithm introduces an equivalence parameter e% and a minimum confidence parameter c%. 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. 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. Reasonable default values set the equivalence parameter e in the range 90–95% and the minimum confidence parameter c in the range 50–70%. In all simulations reported here, parameter values are set apriori to e = 90% and c = 50%. Alternatively, e and c may be chosen by validation.

The following steps derive the list of rules.

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 $\#(x) \leq \#(y)$: 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 $[\#(x\ \&\ y) / \#(y)] \geq e$%. 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 $C \geq e$ 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 $10\% \leq C < c$.

(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 $c \leq [\#(x\ \&\ y) / \#(y)] < e$.

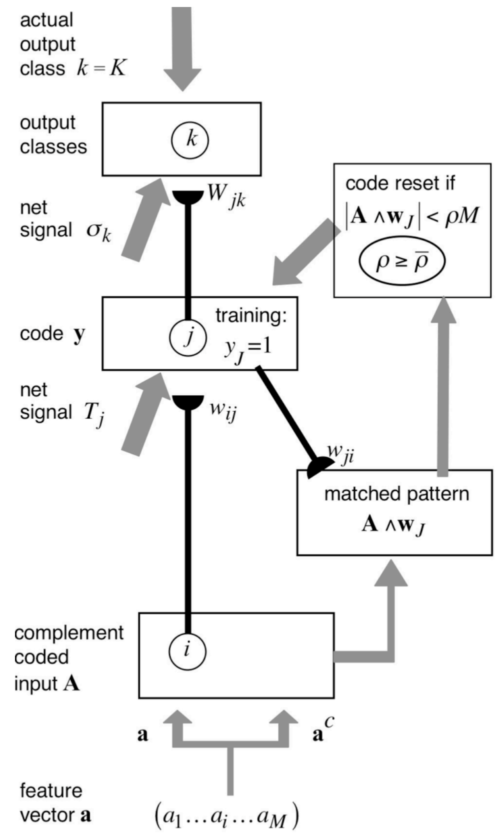

Figure 2 Default ARTMAP notation: An M-dimensional feature vector $\mathbf{a}$ is complement coded to form the 2M-D ARTMAP input $\mathbf{A}$. Vector $\mathbf{y}$ represents a winner-take-all code during training, when a single category node (j=J) is active; and a distributed code during testing. With 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 ARTMAP baseline vigilance matching parameter is set to its default value, $\bar{\rho} = 0$. [21]

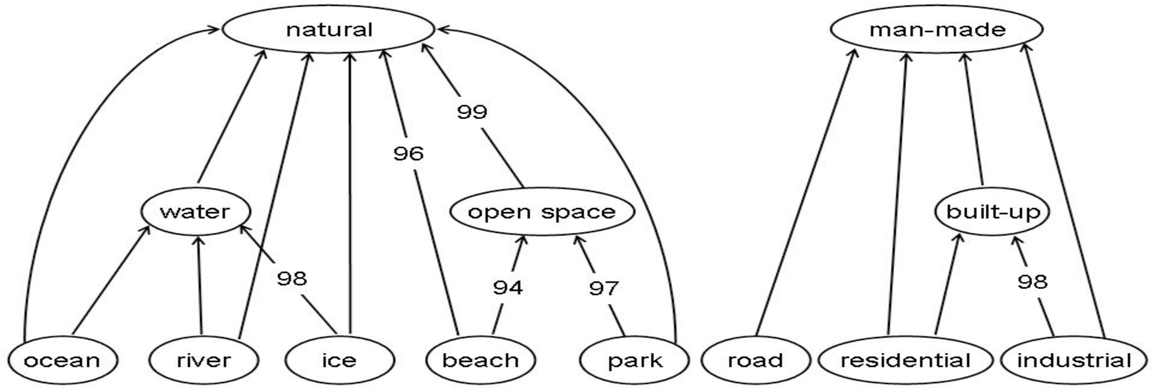

Figure 3 *Man-made/natural* knowledge hierarchy for the Boston testbed using the ARTMAP information fusion system. Numbers on arrows indicate the confidence in the corresponding rule. For arrows with no numbers, C=100%. Parameters 39% < c < 94% and e > 87% all produce the same result.

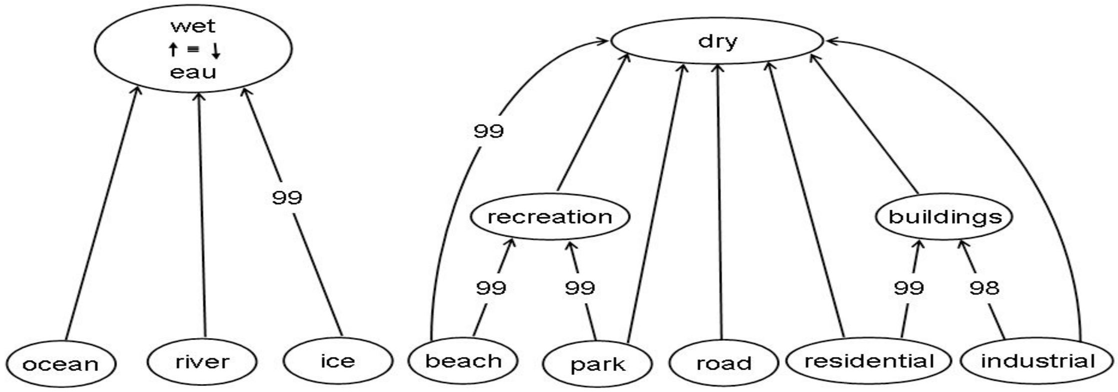

Figure 4 *Wet/dry* knowledge hierarchy obtained as in Figure 3 for the Boston testbed. Parameters 34 % < c < 98% and e > 89% all produce the same result.

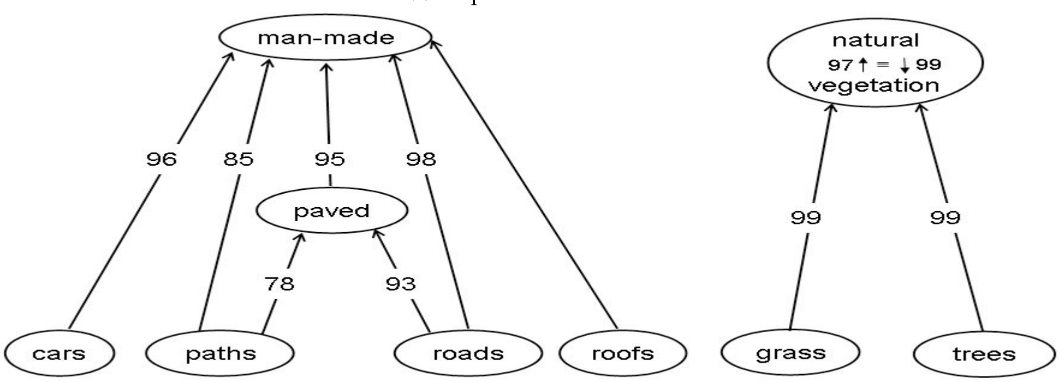

Figure 5 *Man-made/natural* knowledge hierarchy for the Monterey testbed. Parameters 42% < c < 78% and 87% < e < 97% all produce the same result.

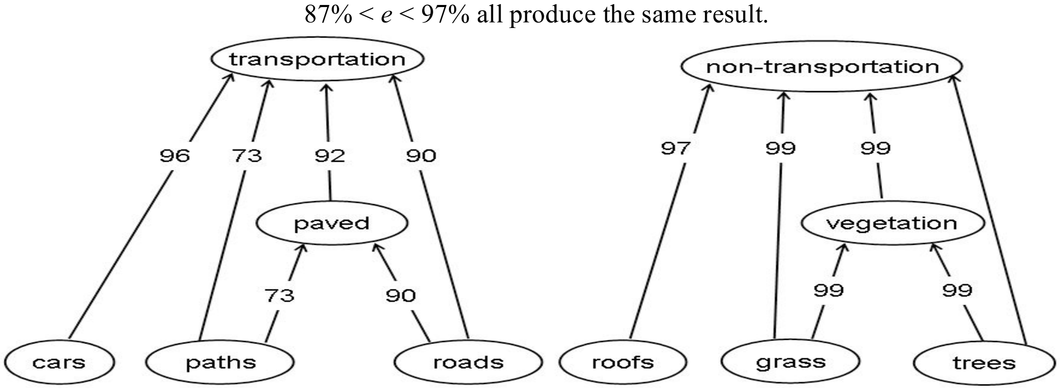

Figure 6 *Transportation/non-transportation* knowledge hierarchy obtained as in Figure 5 for the Monterey testbed. Parameter 47 % < c < 73% all produce the same result.

5 Knowledge domain testbeds

Suppose that a group of experts label portions of the Boston image according to a *man-made* versus *natural* partition of the region. Certain pixels may, for example, be labeled as either *ocean* or *water* or *natural,* and other pixels could be labeled *residential* or *built-up* or *man-made*. Such a labeling scheme defines one knowledge domain.

In an ARTMAP information fusion system, each training pixel is given one of these multiple labels. The relationships among these labels are not specified or even known to the user. The system discovers these relationships and expresses the implicit knowledge of the experts in the form of a hierarchy (Figure 3).

A different group of experts might define a second domain by partitioning the same region as *wet* versus *dry*. A pixel previously labeled *water, ocean*, or *natural* might now be labeled as *ocean, eau,* or *wet*. Rules discovered for such a second knowledge domain are shown in Figure 4.

The ARTMAP system tests cross-domain integration for the Boston domains defined by *man-made* versus *natural* and *wet* versus *dry*. A second example based on the Monterey testbed defines two domains from the image partitions *man-made* versus *natural* and *transportation* versus *non-transportation* (Figures 5, 6).

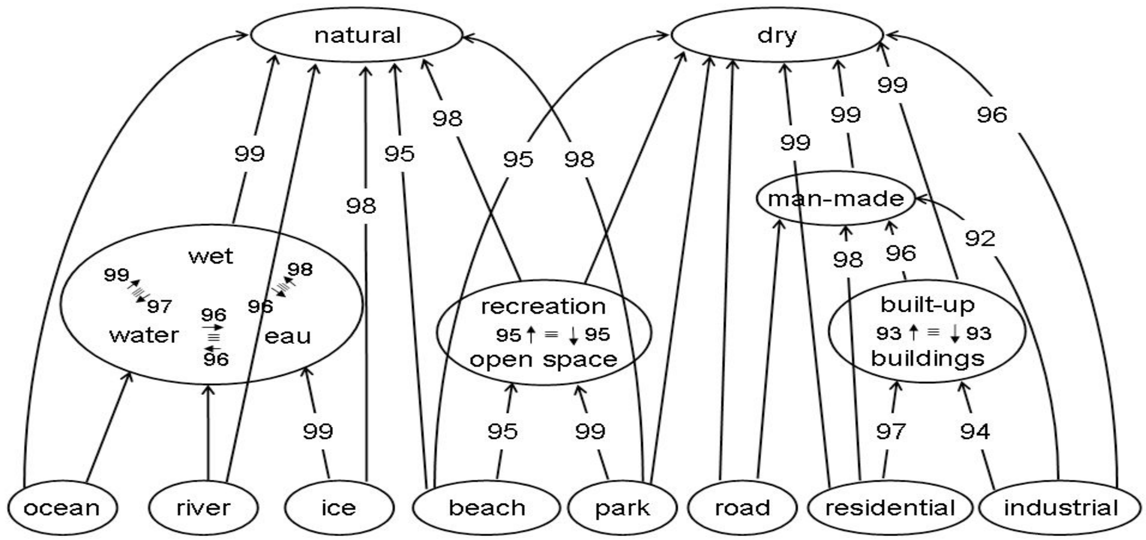

Figure 7 Unified hierarchy obtained on the Boston image using the ARTMAP system showing the fusion of *man-made/ natural* (Figure 3) and *water/dry* (Figure 4) domains. The system correctly discovers all rules as well as the equivalences *open space≡recreation*, *built≡buildings,* and *water≡eau≡wet*. Further, the ARTMAP information fusion system discovers the cross-domain rules *wet*$\Rightarrow$*natural* and *man-made*$\Rightarrow$*dry*. Parameters 25% < *c* < 92% and 82% < *e* < 93% produce the same results.

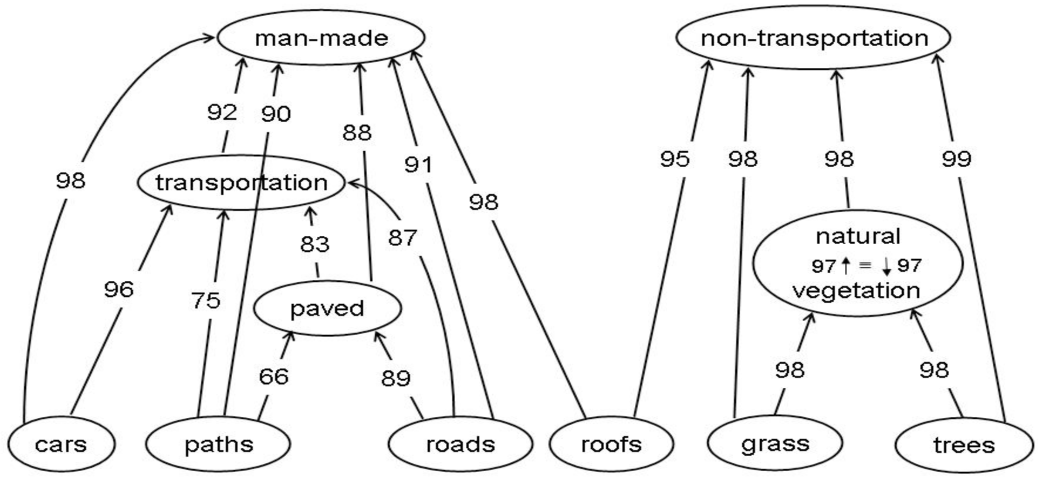

Figure 8 Unified hierarchy obtained on the Monterey image using the ARTMAP system, showing the fusion of *man-made/ natural* (Figure 5) and *transportation/non-transportation* (Figure 6) domains. The system correctly discovers all rules as well as the equivalence *natural≡vegetation*. Further, the ARTMAP information fusion system discovers the cross-domain rules *transportation*$\Rightarrow$*man-made* and *natural*$\Rightarrow$*non-transportation*. Parameters 46% < *c* < 66% and 87% < *e* < 97% produce the same results.

6 Fusion of knowledge domains

When two or more domains have been defined for a given testbed, an ARTMAP fusion system can unify this information as consistent knowledge. Training pixel labels are drawn from both the domains. A pixel over the ocean in the Boston image might be labeled *water* or *natural* (domain 1) or *eau* or *wet* (domain 2) or *ocean* (both domains). Another pixel might be labeled *built-up* or *man-made* (domain 1) or *buildings* or *dry* (domain 2) or *residential* (both domains). In the training set, a pixel is associated with just one such label.

Figure 7 shows the cross-domain knowledge hierarchy discovered for the Boston testbed. The new hierarchy correctly embeds all rules of the individual domains (Figures 3, 4). In addition, the ARTMAP system discovers the cross-domain rules *wet*$\Rightarrow$*natural* and *man-made*$\Rightarrow$*dry*. Note that the top-level classes that partition the two separate domains do not all appear at the top-level of the unified structure.

The fused knowledge domain also defines cross-domain equivalence classes. The class *water* (domain 1) is found to be equivalent to *eau* (domain 2), which is seen, in turn, to be equivalent to *wet* (domain 2). Similarly, *open space* (domain 1) is found to be equivalent to *recreation* (domain 2), and *built-up* (domain 1) to be equivalent to *buildings* (domain 2). Note that the discovery of equivalences uses no explicit domain knowledge. Thus, labels in one domain might be in an unknown language or code.

Figure 8 shows the cross-domain knowledge hierarchy for the Monterey testbed. Included are the cross-domain rules *transportation*$\Rightarrow$*man-made* and *natural*$\Rightarrow$*non-transportation*. The fused hierarchy also correctly includes all rules from the individual domains (Figures 5, 6).

7 Conclusion

The ARTMAP neural network produces one-to-many mappings from input vectors to output classes, as well as the more traditional many-to-one compressed representations of its training space. 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 rules, confidence estimates, and multi-class hierarchical relationships from patterns of distributed test predictions. The system also fuses independent domains into a unified knowledge hierarchy, discovering cross-domain rules in this process. The system infers multi-level and cross-domain rules without any supervised labeling of these relationships.

Acknowledgements

This research was supported by grants from the National Science Foundation (NSF SBE-0354378) and the National Geospatial-Intelligence Agency (NMA 201-01-1-2016).

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, S. Martens, and O.J. Ogas. Self-Organizing information fusion and hierarchical knowledge discovery: a new framework using ARTMAP neural networks. *Neural Networks*, 18(3), 287-295, 2005.

[3] G. A. Carpenter, S. Martens, and O.J. Ogas. Self-organizing hierarchical knowledge discovery by an ARTMAP image fusion system. In *Proceedings of the 7th International Conference on Information Fusion (Fusion 2004)*, 235-242, 2004.

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

[5] 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.

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

[7] 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.

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

[9] 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 Transactions on Neural Networks*, 10(5):1214-1221, 1999.

[10] 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.

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

[12] 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.

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

[14] 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.

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

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

[17] 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.

[18] 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.

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

[20] G.A. Carpenter. Default ARTMAP. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1396-1401, Portland, Oregon, 2003.

[21] G. Amis and G.A. Carpenter. Default ARTMAP 2. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN),* Orlando, Florida, 2007.

[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] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of ACM SIGMOD Conference on Management of Data*, Washington, DC, 1993.