Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31 - August 4, 2005

Self-Organizing Hierarchical Knowledge Discovery by an ARTMAP Information Fusion System

Gail A. Carpenter, Siegfried Martens
Boston University, Department of Cognitive and Neural System
677 Beacon Street, Boston, MA 02215 USA.
[gail, sig] @cns.bu.edu http://cns.bu.edu/techlab/

Abstract – **Classifying terrain or objects may require the resolution of conflicting information from sensors working at different times, locations, and scales, and from users with different goals and situations. Current fusion methods can help resolve such inconsistencies, as when evidence variously suggests that an object is a car, a truck, or an airplane. The methods described here define a complementary approach to the information fusion problem, considering the case where sensors and sources are both nominally inconsistent and reliable, as when evidence suggests that an object is a car, a vehicle, and man-made. Underlying relationships among classes are assumed to be unknown to the automated system or the human user. The ARTMAP self-organizing rule discovery procedure is illustrated with an image example, but is not limited to the image domain.**

I. Introduction

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, such 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 – yet accurate. This is a problem that the human brain solves naturally. A young child who hears the family pet variously called *Spot*, *puppy*, *dog*, *dalmatian*, *mammal*, and *animal* is not only not alarmed by these conflicting 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 sources. Each source labels a portion of the region based on sensor data and observations collected at specific times and based on individual goals and interests. Across sources, 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. Alternatively, an analyst could be faced with complex or unfamiliar labels, or the structure of object relationships may vary from one region to the next.

The current study shows how an ARTMAP neural network can act as a self-organizing expert system to derive hierarchical knowledge structures from nominally 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 classes with a given input. During testing, distributed code activations predict multiple output class labels. A rule production algorithm uses the pattern of distributed network predictions 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.

II. Multi-class Predictions by ARTMAP Neural Networks

While the earliest unsupervised ART [2] and supervised ARTMAP networks [3] feature winner-take-all code representations, many of the networks developed since the mid-1990s 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 [4]. Selection of one particular

0-7803-9048-2/05/$20.00 ©2005 IEEE

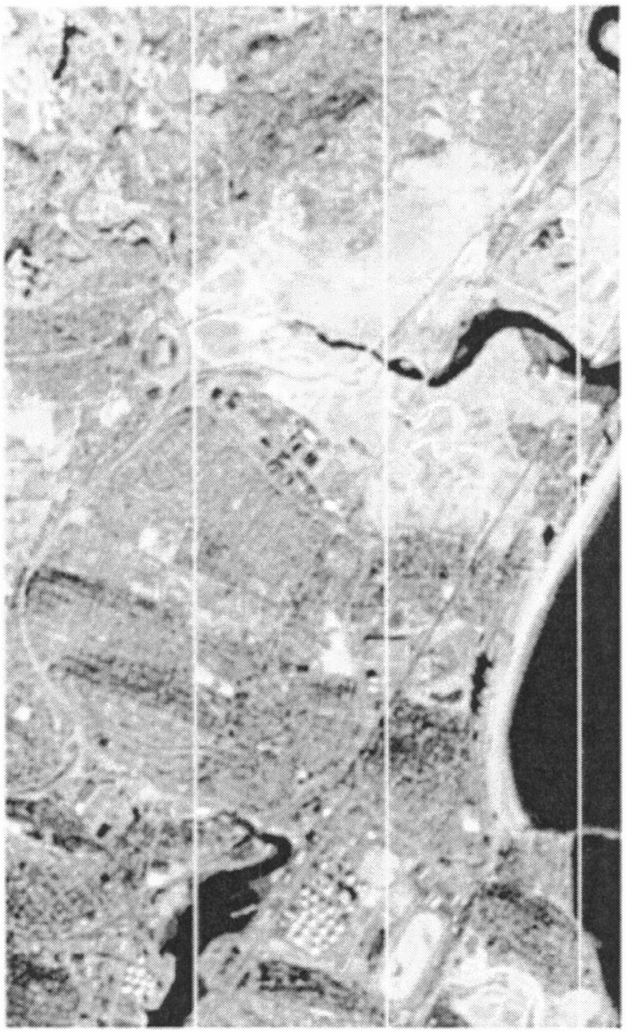

Fig. 1. Testbed Boston image for ARTMAP information fusion methods, in grey scale representation of preprocessed inputs. The city of Revere is at the center, surrounded by (clockwise from lower right) portions of Winthrop, East Boston, Chelsea, Everett, Malden, Melrose, Saugus, and Lynn. Logan Airport 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. Dimensions: 360 x 600 pixels (15*m* resolution) $\cong$ 5.4 *km* x 9 *km*. The image is divided into four vertical strips: two for training, one for validation (if needed), and one for testing. This protocol produces geographically distinct training and testing areas, to assess regional generalization. Typically, class label distributions vary substantially across strips.

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 another one that selects class-specific output thresholds via validation [5]. Distributed coding supports each method, but the ultimate prediction is one output class per test input. This procedure 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 separate locations. This separation tests for generalization to new regions, where output class distributions could typically be far from those of the training and validation sets.

The information fusion techniques developed in the current study modify the baseline mapping procedure by allowing the system to predict more than one output class during testing. A given test pixel either predicts the N classes receiving the largest net system outputs or predicts all classes whose net output exceeds a designated threshold Γ. A preliminary version of the ARTMAP information fusion system [6] chose a global selection parameter N or Γ based on analysis of the validation strip. This method succeeds when most validation and test items share a common number of correct output classes. The preferable procedure used here allows each test exemplar to choose its own number N of output class predictions. This per-pixel filtering method thus does not rely on the strong assumption that the correct number of output classes per item is approximately uniform across the test set.

An image testbed demonstrates the robustness of the ARTMAP information fusion procedure. This example was derived from a Landsat 7 Thematic Mapper (TM) data acquired on the morning of January 1, 2001 by the Earth Resources Observation System (EROS) Data Center, U.S. Geological Survey, Sioux Falls, SD (http://edc.usgs.gov). The area includes portions of northeast Boston and suburbs (Fig. 1), and encompasses mixed urban, suburban, industrial, water, and park spaces. Ground truth pixels are labeled *ocean*, *ice*, *river*, *beach*, *park*, *road*, *residential*, *industrial*, *water*, *open space*, *built-up*, *natural*, *man-made*. During training, ARTMAP is given no information about relationships among the target classes.

III. Deriving a Knowledge Hierarchy from a Trained Network: Predictions, Rules, and Graphs

The *ARTMAP fusion system* provides a canonical procedure for assigning to each input an arbitrary number of output classes in a supervised learning setting. Information implicit in the distributed predictions of a trained ARTMAP network, trained with prescribed protocols [7], can be used to generate a hierarchy of output class relationships. To accomplish this, each test pixel first produces a set of output class predictions. The resulting list of test predictions determines a list of rules $x \Rightarrow y$ which define relationships between pairs of output classes, with each rule carrying a confidence value. The rules are then used to assign classes to levels, with rule antecedents x at lower levels and consequents y at higher levels. Classes connected by arrows that codify the list of rules and confidence values form a graphical representation of the knowledge hierarchy, as follows.

A. Predictions

A critical aspect of the default ARTMAP network (Fig. 2) 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. 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 employed here 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., *road*, *pavement*, *man-made*), 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 signal, and the per-pixel filtering method would predict three classes.

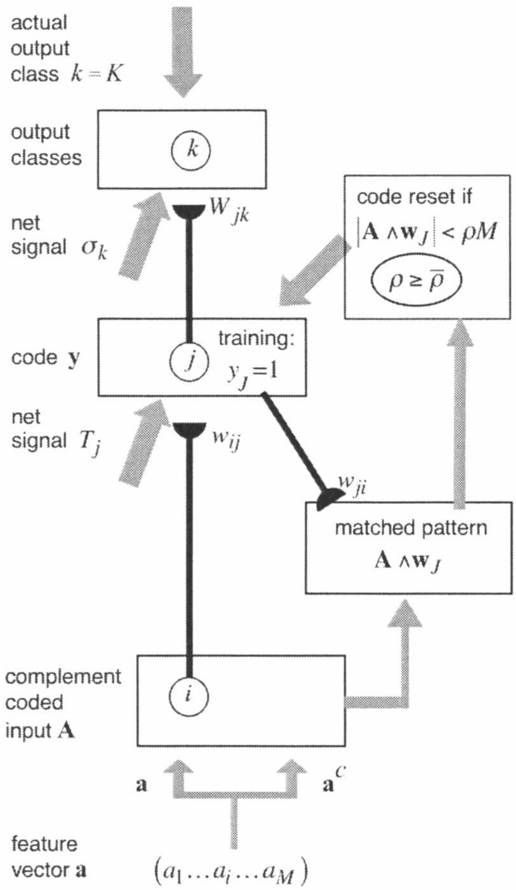

Fig. 2. Default ARTMAP notation: An M-dimensional feature vector **a** is complement coded to form the 2M-D ARTMAP input **A**. Vector **y** represents a winner-take-all code during training, when a single category node (j=J) is active; and a distributed code during testing. With fast learning, bottom-up weights w_{ij} equal top-down weights w_{ji}, and the weight vector $\mathbf{w}_j$ represents their common values. When a coding node j is first selected during training, it is connected to the output class k of the current input ($W_{jk} = 1$). During testing, a distributed code **y** produces predictions σ_k distributed across output classes. In all simulations reported here, the baseline vigilance matching parameter $\bar{\rho}$=0. [4]

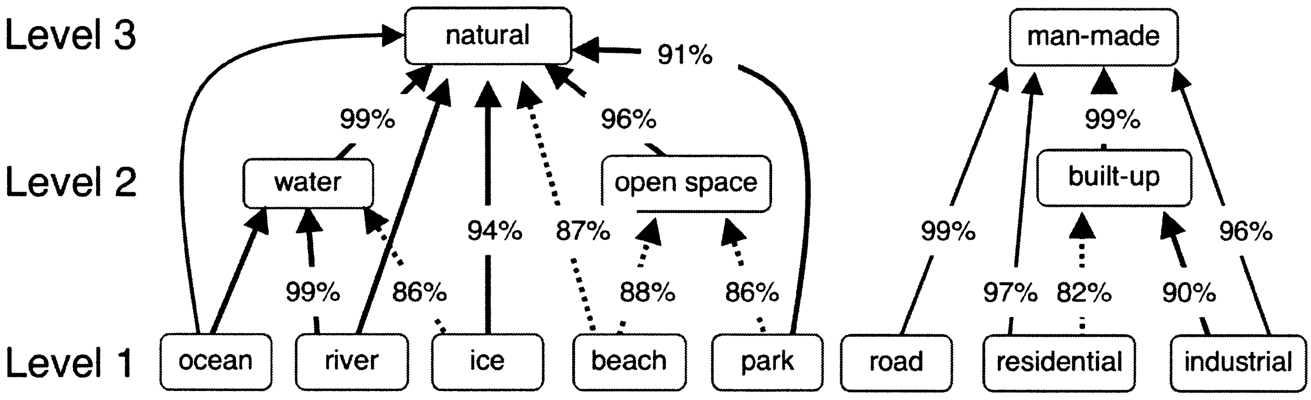

Fig. 3. For the Boston example, the ARTMAP fusion system correctly produces all class rules and levels. Each rule's confidence, if less than 100%, is printed on its arrow. Dashed arrows indicate rules with confidence below 90%.

B. Rules

Once each test pixel has produced a set of output class predictions $\{x,y,\ldots\}$ from its distributed signals σ_k, 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 $x \Longrightarrow y$, each carrying a confidence value C%. This rule creation method is related to the Apriori algorithm in the association rule literature [8, 9].

The five steps listed below produce the list of rules that label class relationships. The algorithm employs 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. Although this "no extinction" clause may produce low-confidence rules, these may occasionally correspond to cases that are rare but important. The user can easily take these exceptions under advisement, since the summary graph displays each confidence value. Two classes x and y are treated as *equivalent* ($x \equiv y$) if both rules $x \Longrightarrow y$ and $y \Longrightarrow x$ 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 were set *a priori* to e=90% and c=50%. Alternatively, e and c may be chosen by validation.

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.

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 $\#(x) \geq \#(y)$: classes x observe the order established in Step 1; and for each such class x, classes y observe the same order.

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 (where $\#(x) \geq \#(y)$, as in Step 2).

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

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 $x \Longrightarrow y$ with maximal $\#(x\,\&\,y)$ pixels.

C. Graphs

A directed graph summarizes the list of implication rules. These rules suggest a natural hierarchy among output classes, with antecedents sitting below consequents. For each rule $x \Longrightarrow y$, 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 $x \Longrightarrow y$ produces an arrow from x to y. Each rule's confidence is indicated on the arrow, with lower-confidence rules (say C<90%) having dashed arrows. For arrows with no displayed confidence values, C=100%.

The following procedure assigns 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.

The graph in Fig. 3 depicts the implication rules, hierarchy levels, and confidence values derived for the Boston example. ARTMAP information fusion has placed each class in its correct level and discovers all the correct rules.

IV. 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 mappings, as the normal product of its supervised learning laws. During training, a given input may learn associations to 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 specification of the rules underlying these relationships. One-to-many learning allows the ARTMAP information fusion system to associate any number of output classes with each input. Although inter-class information is not given 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 Boston image testbed example 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, however, limited to the image domain illustrated here, and could be applied, for example, to infer patterns of drug resistance or to improve marketing suggestions to individual consumers. One such pilot study has created a hypothetical set of relationships among protease inhibitors, based on resistance patterns from genome sequences of HIV patients.

Acknowledgments

This work was supported by research grants from the Air Force Office of Scientific Research (AFOSR F49620-01-1-0423), the National Geospatial-Intelligence Agency (NMA 201-01-1-2016), the National Science Foundation (NSF SBE-0354378), and the Office of Naval Research (ONR N00014-01-1-0624); and by postdoctoral fellowships from the National Geospatial-Intelligence Agency and the National Science Foundation for Siegfried Martens (NMA 501-03-1-2030 and NSF DGE-0221680).

References

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

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

[3] Carpenter, G.A. Grossberg, S., & 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.

[4] Carpenter, G.A. (2003). Default ARTMAP. In *Proceedings of the international joint conference on neural networks (IJCNN'03)*, Portland, Oregon (pp. 1396-1401).

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

[6] Carpenter, G.A., Martens, S., & Ogas, O.J. (2004). Self-organizing hierarchical knowledge discovery by an ARTMAP image fusion system. In *Proceedings of the 7th international conference on information fusion, Stockholm, Sweden* (pp. 235-242).

[7] Carpenter, G.A., Martens, S., & Ogas, O.J. (2005). Self-organizing information fusion and hierarchical knowledge discovery: a new framework using ARTMAP neural networks. *Neural Networks*, **18**.

[8] Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the international conference on management of data (ACM SIGMOD)*, Washington, DC (pp. 207-216).

[9] Agrawal, R. & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th international conference on very large data bases (VLDB)*, Santiago, Chile (pp. 487-499).