Scalable Model for Extensional and Intensional Descriptions of Unclassified Data

Hércules A. Prado[1,2,⋆], Stephen C. Hirtle[2,⋆⋆], Paulo M. Engel[1,⋆⋆⋆]

[1] Universidade Federal do Rio Grande do Sul
Instituto de Informática
Av. Bento Gonçalves, 9500 - Bairro Agronomia
Porto Alegre / RS - Brasil
Caixa Postal 15.064 - CEP 91.501-970
Fone: +55(051)316-6829
Fax: +55(051)319-1576
[2] University of Pittsburgh
Department of Information Sciences and Telecommunications
135 North Bellefield Ave. Pittsburgh, PA 15.260
Phone: +1(412)624-9434
Fax: +1(412)624-2788
{prado, engel}@inf.ufrgs.br and hirtle+@pitt.edu

Abstract. Knowledge discovery from unlabeled data comprises two main tasks: identification of "natural groups" and analysis of these groups in order to interpret their meaning. These tasks are accomplished by unsupervised and supervised learning, respectively, and correspond to the taxonomy and explanation phases of the discovery process described by Langley [9]. The efforts of Knowledge Discovery from Databases (KDD) research field has addressed these two processes into two main dimensions: (1) scaling up the learning algorithms to very large databases, and (2) improving the efficiency of the knowledge discovery process. In this paper we argue that the advances achieved in scaling up supervised and unsupervised learning algorithms allow us to combine these two processes in just one model, providing extensional (who belongs to each group) and intensional (what features best describe each group) descriptions of unlabeled data. To explore this idea we present an artificial neural network (ANN) architecture, using as building blocks two well-know models: the ART1 network, from the Adaptive Resonance Theory family of ANNs [4], and the Combinatorial Neural Model (CNM), proposed by Machado ([11] and [12]). Both models satisfy one important desiderata for data mining, learning in just one pass of the database. Moreover, CNM, the intensional part of the architecture, allows one to obtain rules directly from its structure. These rules represent the insights on the groups. The architecture can be extended to other supervised/unsupervised learning algorithms that comply with the same desiderata.

⋆ Researcher at EMBRAPA — Brazilian Enterprise for Agricultural Research and lecturer at Catholic University of Brasília (Supported by CAPES - Coordenaçao de Aperfeiçoamento de Pessoal de Nível Superior, grant nr. BEX1041/98-3)
⋆⋆ Professor at University of Pittsburgh
⋆⋆⋆ Professor at Federal University of Rio Grande do Sul

1 Introduction

Research in Knowledge Discovery from Databases (KDD) has developed along two main dimensions: (1) improving the knowledge discovery process, and (2) scaling up this same process to very large databases. Machine Learning, as an important field related to KDD, is founded on three principles [19]:

1. Modeling of cognitive processes, aiming to select characteristics of interest to be formalized as knowledge;
2. Computer science, which offers a formalism to support the descriptions of those characteristics, as well as providing approaches to evaluate the degree of computational difficulty of the issues involved; and
3. Applications, where one departs from practical needs to the implementation of systems.

In this article, we depart from a characterization of the concept formation activity as a cognitive process, proposing a computational approach to support this activity, from the point of view of KDD. We take the concept of performance as given by the relation functionality/resources applied. By this way, we present a model where funcionality is increased while the resources applied are just slightly changed.

2 Motivation

According to Wrobel [19], a concept is ”a generalized description of sets of objects”. In this sense, Easterlin and Langley [5] analyse the concept formation process as follows:

1. Given a set of objects (instances, events, cases) descriptions, usually presented incrementally;
2. find sets of objects that can be grouped together (aggregation), and
3. find intensional description of these sets of objects (characterization).

Murphy and Medin [14] discuss two hypothesis that constrain the way objects are grouped in concepts:

1. Similarity hypothesis: this hypothesis sustain that what defines a class is that its members are similar to each other and not similar to members of other classes.
2. Correlated attribute hypothesis: this hypothesis states that ”natural groups” are described according to clusters of features and that categories reflect the cluster structure of correlations.

The first hypothesis presents a problem that is: the similarity criteria must be applied to a pre-defined set of features and the definition of this set is affected by the previous knowledge one has over the objects. However, when just a small knowledge about the data exists, this criteria is used as a first approximation. Over this approximation, the correlated attribute hypothesis is applied. In a broad sense, what is desirable in this process is to provoke the mental operations that can lead to a problem solution ([16] and [15]). Actually, since it seems that the discovery process, as a rule, requires the human judgment [9], it is useful to leave available to the analyst all relevant information to evaluate both hypothesis when searching for the classes' structures.

3 Proposed Architecture

The research on the KDD realm has emphasized improvements in the processes of supervised and unsupervised learning. More recently, many unsupervised learning algorithms have been scaled up according to the desiderata proposed by Agrawal *et al.* [1] for this kind of learning algorithm. Considering these advances and the ones in supervised learning, we believe there is enough room to scale up the combined process of unsupervised and supervised learning in order to obtain better descriptions of unclassified data. By ”better descriptions” we mean obtaining intensional (what are the main characteristics of each class) descriptions, beyond the extensional (what objects are members of each class) ones, usually provided. We explore our idea with a hybrid architecture, based into two well-know models: ART1 [4], used for cluster binary data, and CNM ([11] and [12]), used to map the input space in the formed classes. Both ANNs present an important characteristic to support data mining: they learn in just one pass of the entire data set. The model is illustrated by Figure 1.

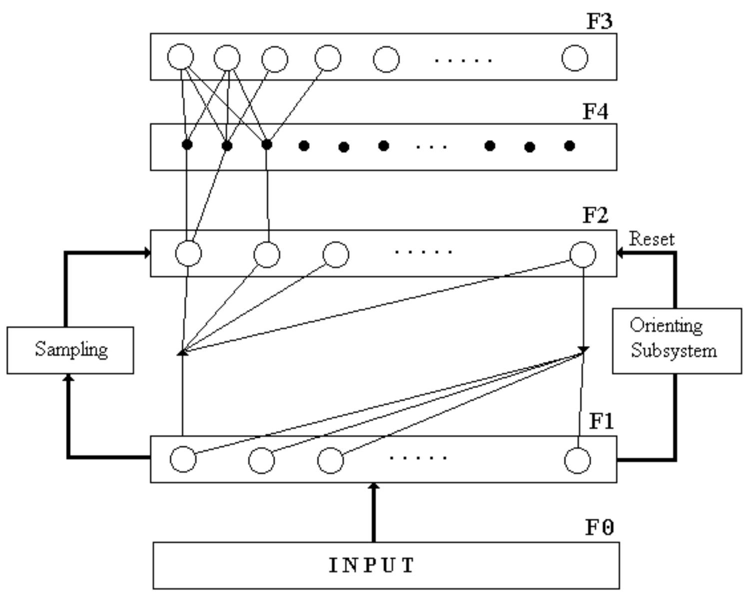

Fig. 1. Describing unclassified data

The architecture is composed by five layers according to the schema: Input layer (F0, where the examples are introduced in the architecure); Aggregation module (F1 and F2, where the classes are defined); Characterization module (F2, F3 , and F4, where the classes are explained).

For the characterization module, it requires the pre-existence of classes that would not be available when the process starts. We overcome this problem by creating the classes by means of a sampling subsystem. The creation of classes through sampling was explored by Guha [8] with consistent results. As a consequence of the sampling process, a complete execution of the system will take more than one pass of the input data set. Considering γ the size of the sample and **D** the size of the input data set, a complete execution of the system will take, precisely, $\mathbf{D} + \gamma$ records. In the next two sections, we describe each model used as building blocks for our architecture.

4 ART1 Neural Network

ART1 (F1 and F2 layers) is a member of the so-called ART family, that stands by Adaptive Resonance Theory [10], [3], [4] and [6].

ART1 is a competitive recurrent network with two layers, the input layer and the clustering layer. This network was developed to overcome the plasticity-stability dilemma [7], allowing an incremental learning, with a continuous updating of the clusters prototypes, and preserving the previously stored patterns. The clustering algorithm proceeds, in general steps, as folows: (a) the first input is selected to be the first cluster; (b) each next input is compared with each existing cluster; the first cluster where the distance to the input is less then a threshold is chosen to cluster the input. Otherwise, the input defines a new cluster. It can be observed that the number of clusters depends on the threshold and the distance metric used to compare the inputs with the clusters. For each input pattern presented to the network, one output unit is declared winner (at the first pattern, the own input pattern defines the cluster). The winner backpropagates a signal that encodes the expected pattern template. If the current input pattern differs more than a defined threshold from the backpropagated signal, the winner are temporarily disabled (by the Orienting System) and the next closest unit is declared winner. The process continues until an output unit become a winner, considering the threshold. If no one of the output units become a winner, a new output unit is defined to cluster the input pattern. Graphically, an ART1 network can be illustrated by Figure 2, where it appears with four input and six ouput neurons. t_{ij} and b_{ij} are, respectively, bottom-up and top-down connections.

One important characteristic of this ANN is that it works in just one pass, what is interesting when we are processing a huge amount of data. The training algorithm for this network is the following:

- **Step1. Initialization**: The bottom-up $b_{ij}(t)$ and top-down $t_{ij}(t)$ weight connection between input node i and output node j at time t are set up. The fraction ρ (vigilance parameter) is defined, indicating how close an input must be to a stored exemplar to match. Initialize N and M, numbers of input and output nodes.
- **Step 2. Apply New Input**
- **Step 3. Compute Matching Scores**: The bottom-up weights are applied to the input pattern, generating the output signal: μ_j.
- **Step 4. Select Best Matching Exemplar**: $\mu_j^* = \max\{\mu_j\}$ is taken as the best exemplar.
- **Step 5. Vigilance Test**: The best exemplar and the input pattern are compared, according to ρ. If the distance is acceptable, the control flows to **Step 7**, otherwise **Step 6** proceeds.

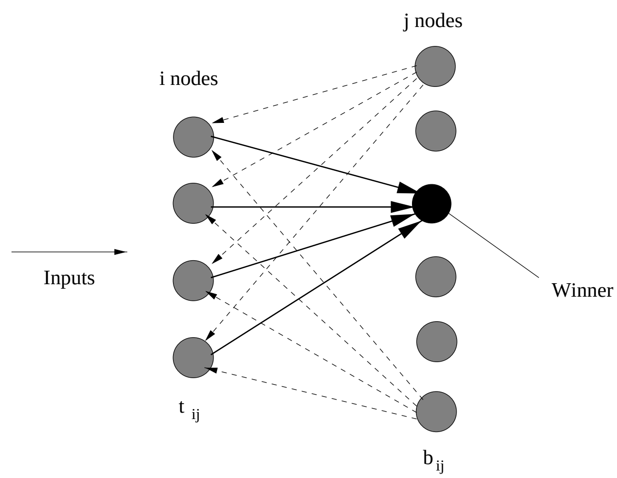

Fig. 2. Architecture of an ART network [3]

- **Step 6. Disable Best Matching Exemplar**: The ouput of the best matching node selected in Step4 is temporarily set to zero and no longer takes part in the maximization of Step4. Then go to Step 3.
- **Step 7. Adapt Best Matching Exemplar**:

$$t_{ij_*}(t+1) = t_{ij_*}(t)t(x_i)$$
$$b_{ij_*}(t+1) = \frac{t_{ij_*}(t)x_i}{\frac{1}{2} + \sum_{i=0}^{N-1} t_{ij_*}(t)x_i}$$

- **Step 8. Repeat by Going to Step 2**: First enable any node disabled in Step 6.

5 Combinatorial Neural Model (CNM)

CNM (F2, F3, and F4 layers) is a hybrid architecture for intelligent systems that integrates symbolic and connectionist computational paradigms. This model is able to recognize regularities from high-dimensional symbolic data, performing mappings from this input space to a lower dimensional output space. Like ART1, this ANN also overcomes the plasticity-stability dilemma [7].

The CNM uses supervised learning and a feedforward topology with: one input layer, one hidden layer - here called combinatorial - and one output layer (Figure 3). Each neuron in the input layer corresponds to a concept - a complete idea about an object of the domain, expressed in an object-attribute-value form. They represent the evidences of the domain application. On the combinatorial layer there are aggregative fuzzy AND neurons, each one connected to one or more neurons of the input layer by arcs with adjustable weights. The output layer contains one aggregative fuzzy OR neuron for each possible class (also called hypothesis), linked to one or more neurons on the combinatorial layer. The synapses may be excitatory or inhibitory and they are characterized by a strength value (weight) between zero (not connected) to one (fully connected synapses).

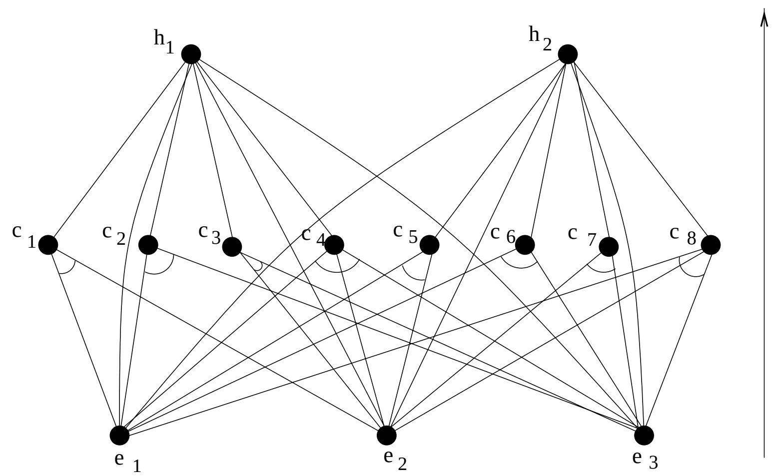

Fig. 3. Complete version of CNM for 3 input evidences and 2 hypotheses [11]

The network is created completely uncommited, according to the following steps: (a) one neuron in the input layer for each evidence in the training set; (b) a neuron in the output layer for each class in the training set; and (c) for each neuron in the output layer, there is a complete set of hidden neurons in the combinatorial layer which corresponds to all possible combinations (length between two and nine) of connections with the input layer. There is no neuron in the combinatorial layer for single connections. In this case, input neurons are connected directly to the hypotheses.

The learning mechanism works in only one iteration, and it is described bellow:

PUNISHMENT_AND_REWARD_LEARNING_RULE

– **Set** to each arc of the network an accumulator with initial value zero;
– **For each** example case from the training data base, **do**:
  - *Propagate* the evidence beliefs from input nodes until the hypotheses layer;
  - **For each** arc reaching a hypothesis node, **do**:
    - ∗ **If** the reached hypothesis node corresponds to the correct class of the case
    - ∗ **Then** *backpropagate* from this node until input nodes, increasing the accumulator of each traversed arc by its evidencial flow (Reward)
    - ∗ **Else** *backpropagate* from the hypothesis node until input nodes, decreasing the accumulator of each traversed arc by its evidencial flow (Punishment).

After training, the value of accumulators associated to each arc arriving to the output layer will be between [-T, T], where T is the number of cases present in the training set. The last step is the prunning of network; it is performed by the following actions: (a) remove all arcs whose accumulator is lower than a threshold (specified by a specialist); (b) remove all neurons from the input and combinatorial layers that became disconnected from all hypotheses in the output layer; and (c) make weights of the arcs arriving at the output layer equal to the value obtained by dividing the arc accumulators by the largest arc accumulator value in the network. After this pruning, the network becomes operational for classification tasks. This ANN has been applied with success in data mining tasks ([2], [17], and [18]).

6 Ongoing Work

This paper presents an architecture to scale up the whole process of concept formation according two main constraints: the identification of groups composed by similar objects and the description of this groups by the higher correlated features. The ongoing work includes the implementation and evaluation of this architecture, instantiated to ART1 and CNM, and its extension to cope with continuous data.

References

1. Agrawal, R., Gehrke, J., Gunopulos, D., Raghanavan, P. Automatic Subspace Clustering of High-Dimensional Data for Data Mining Applicati ons. In: Proceedings of ACM SIGMOD98 International Conference on Management of Data, Seattle, Washington, 1998.
2. Beckenkamp, F. G., Feldens, M. A., Pree, W.: Optimizations of the Combinatorial Neural Model. IN: Vth Brazilian Symposium on Neural Networks. (SBRN' 98), Belo Horizonte, Brazil.
3. Bigus, J. P. Data Mining with Neural Networks. [S.l.]: McGraw-Hill, 1996. p.3-42.
4. Carpenter, G. and Grossberg, S. Neural Dynamics of Category Learning and Recognition: Attention, Memory, Consolidation, and Amnesia. In: Joel L. Davis (ed.), Brain structure, learning, and memory. AAAS Symposia Series, Boulder, CO: Westview Press, 1988. p.233-287.
5. Easterlin, J.D., Langley, P.: A Framework for Concept Formation. In: Seventh Annual Conference of the Cognitive Science Society, Irvine, CA, 1985.
6. Engel, P. M. Lecture Notes. Universidade Federal do Rio Grande do Sul. Porto Alegre-RS, Brazil: CPGCC da UFRGS, 1997.
7. Freeman, J. A., Skapura, D. M.: Neural Networks, Algorithms, Applications, and Program Techniques. [S.l.]: Addison-Wesley, 1992. p.292-339.
8. Guha, S., Rastogi, R., Shim, K. Cure: An Efficient Clustering Algorithm for Larg e Databases. In: Proceedings of ACM SIGMOD98 International Conference on Managemen t of Data, Seattle, Washington, 1998.
9. Langley, P. The Computer-Aided Discovery of Scientific Knowledge. In: Proc. of the First International Conference on Discovery Science, Fukuoka, Japan, 1998.
10. Lippmann, D. An Introduction to Computing with Neural Nets, IEEE ASSP Magazine. April, 1987.
11. Machado, R. J., Rocha, A. F.: Handling knowledge in high order neural networks: the combinatorial neural network. Rio de Janeiro: IBM Rio Scientific Center, Brazil, 1989. (Technical Report CCR076).
12. Machado, R. J., Carneiro, W., Neves, P. A.: Learning in the combinatorial neural model, IEEE Transactions on Neural Networks, v.9, p.831-847. Sep.1998.
13. Medin, D., Altom, M.W., Edelson, S.M. and Freko, D. Correlated symptoms and simulated medical classification. Journal of Experimental Psychology: Learning, Memory and Cognition, 8:37-50, 1983.
14. Murphy, G. and Medin, D.: The Role of Theories in Conceptual Coherence. Psychological Review, 92(3):289-316, July, 1985.

15. Pereira, W. C. de A. Resoluçao de Problemas Criativos: Ativaçao da Capacidade de Pensar. Departamento de Informaçao e Documentaçao/EMBRAPA, Brasília-DF, 1980. 54pp.
16. Polya, G. How to Solve It: A New Aspect of Mathematical Method. Princeton: Princeton University Press, 1972. 253pp.
17. Prado, H. A., Frigeri, S. R., Engel, P. M.: A Parsimonious Generation of Combinatorial Neural Model. IN: IV Congreso Argentino de Ciencias de la Computación (CACIC'98), Neuquén, Argentina, 1998.
18. Prado, H. A. do; Machado, K.F.; Frigeri, S. R.; Engel, P. M. Accuracy Tuning in Combinatorial Neural Model. PAKDD'99 - Pacific-Asia Conference on Knowledge Discovery and Data Mining. Proceedings ... Beijing, China, 1999
19. Wrobel, S. Concept Formation and Knowledge Revision. Dordrecht, The Netherlands: Kluwer, 1994. 240pp.