

Generating clusters' explanations in just one data scan

H. A. do Prado^{1,2}, P. M. Engel³ & A. Haëndchen Filho⁴

¹*Graduate Program on Knowledge and TI Management, UCB – Catholic University of Brasilia, Brazil*

²*Center for Agricultural Research on Savannah, Embrapa – Brazilian Agricultural Research Corporation, Brazil*

³*Informatics' Institute, UFRGS – Federal University of Rio Grande do Sul, Brazil*

⁴*Informatics' Department, PUC-Rio - Pontifical Catholic University of Rio de Janeiro, Brazil*

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 phases of the discovery process described by Langley. The efforts of Knowledge Discovery from Databases (KDD) research field have 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 KDD 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 stream, providing extensional and intensional descriptions of unlabeled data. We propose a framework, called Orpheo, which enables the integration of any two unsupervised and supervised algorithms to compose the stream. A particular advantage of our approach is that, if the two algorithms are incremental, the system will build the model in just one data scan. This characteristic satisfies important desiderata for clustering in data mining posed by Bradley *et al.* The framework is instantiated using as building blocks two incremental neural

networks: the ART1 model and the Combinatorial Neural Model. An application related to the agricultural research is presented to illustrate the approach.

1 Introduction

Knowledge discovery from unlabeled data comprises two main tasks: identification of “natural groups” and analysis of these groups in order to interpret their meaning. Research on Knowledge Discovery from Databases (KDD) has addressed these two processes into two main dimensions: (1) scaling up the learning algorithms to very large databases and (2) making easier the whole process of knowledge discovery. The second dimension refers to tasks like data gathering and cleaning, parameter and feature selection, and so on. In this research we argue that the advances achieved in unsupervised and supervised learning under the flag of KDD allow us to combine these two processes in just one stream, providing extensional and intensional descriptions of unlabeled data. Extensional description means to find out which data points belong to each cluster and intensional description corresponds to specifying which features are more correlated to each cluster. To explore this idea we present a framework to integrate different unsupervised and supervised learning algorithms.

2 Motivation

Langley [10] states that, “perhaps, the earliest discovery activity involves the formation of *taxonomies* ... which groups entities as categories and subcategories in a hierarchical manner.” After that, scientists need to discover qualitative laws that characterize the behaviour of those taxonomies or that relate them to each other. The concept formation process, according to Easterlin and Langley [5] is accomplished by the following steps: (1) Given a set of object descriptions, usually presented incrementally; (2) Find sets of objects that can be grouped together (aggregation); and (3) Find intensional descriptions of these sets of objects (characterization).

Murphy and Medin [13] discuss two hypotheses that constrain the way objects are grouped in concepts: (1) the application of an unsupervised learning algorithm to identify clusters; and (2) application of a supervised algorithm to search the correlation structures inside the clusters. Usually this is done applying separately the two kinds of algorithms. Considering the advances in KDD, we propose Orpheo, a framework that integrates different algorithms to improve this whole process. By means of this framework, while searching for knowledge in non-classified data one can find automatically, both the clusters and their explanations on the basis of the most correlated features.

3 Framework description

Orpheo, as shown in Figure 1, consists of three layers: interface, logical, and persistency. The classes names belonging to each layer starts with IL, LL, and PL, respectively. The separation in layers is intended to provide more flexibility while assuring functional independence.

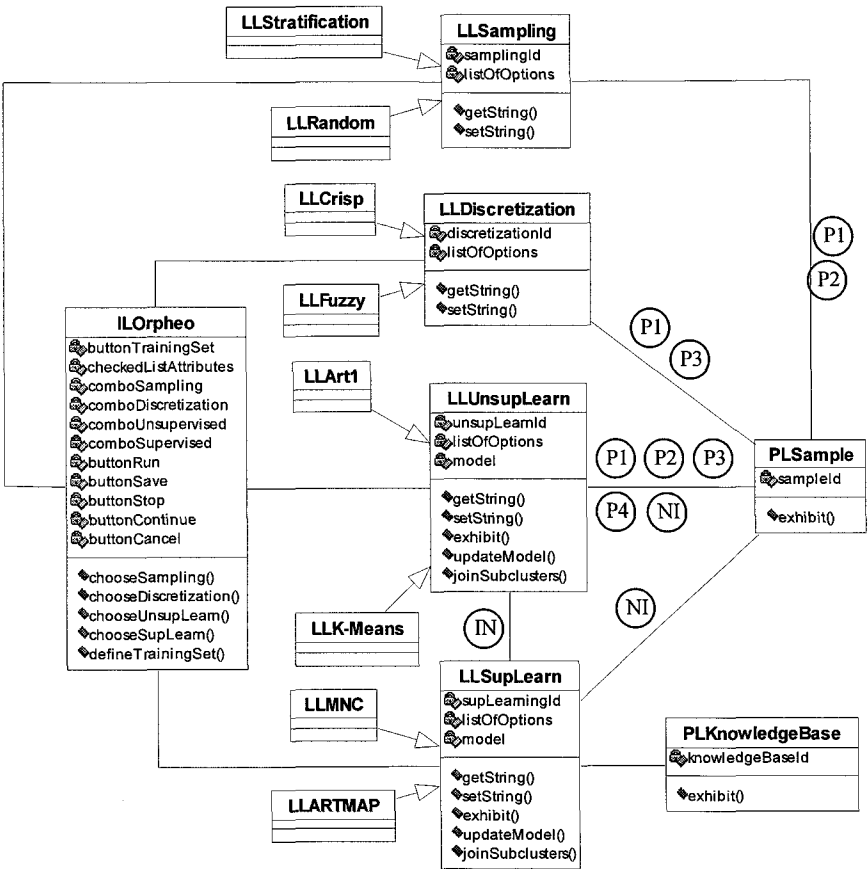


Figure 1. Orpheo framework

The interface layer manages all the communication between the user and the logical layer. Through this layer, the user states her (his) options regarding to sampling, discretization, unsupervised and supervised learning, beyond specifying the training set and the parameters required by each model. The methods starting with the word “choose” encompass *case* structures that control the logical layer flow, according to the user choices. The logical layer gathers the abstract classes for performing those choices and keeps the list of options for execution. All classes in the logical layer are provided with data access services.

Moreover, the classes for learning have services for exhibiting and updating the model. A service for joining clusters is specified for those extensions that applies, e.g., the framework of Bradley *et al.* [3] which works with subclusters that, eventually, join to the clusters. The classes in the persistency layer provide the services for keeping data and knowledge, and have a standard method for exhibiting their structures. Note that **PLSample** is a proper subset of the training set. The flow between logical and persistency layers are also governed by user choices. There are two kinds of flows, one that depends on some pre-processing and are identified with the starting letter P, and other that depends on if at least one of the learning algorithms is incremental or not (identified by the labels IN and NI, respectively). The flow P1 is active when the user chooses both sampling and discretization options, P2 corresponds to the sampling choice, P3 works for the discretization case, and P4 is activated when neither sampling nor discretization is chosen. This framework provides the necessary flexibility to support the user in deciding about which algorithm to apply for each function.

4 Instantiation of the framework

In order to have practical evaluations of the framework, we instantiated it with two artificial neural networks (ANNs): the ART1 network (Figure 2), from the Adaptive Resonance Theory family of ANNs [4], and the Combinatorial Neural Model – CNM (Figure 4), proposed by Machado ([11] and [12]). Both models accomplish some important *desiderata* [3] for data mining, learning in just one pass of the database and being stoppable at any time issuing a working model. Moreover, CNM, the intensional part of the architecture, allows one to obtain rules directly from its representation.

4.1 ART1 model

ART1 is a member of the so-called ART family, which stands for Adaptive Resonance Theory ([4], [6] and [9]). ART1 is a competitive recurrent ANN with two layers: the input layer and the clustering layer. This network was developed to overcome the plasticity-stability dilemma [4], allowing an incremental learning, with a continuous updating of the cluster prototypes, and preserving the previously stored patterns. The clustering algorithm proceeds, in general steps, as follows: (a) the first input is selected to be the first cluster; (b) each next input is compared with each existing cluster; the nearest cluster where the distance to the input is less than 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 input with the clusters. For each input pattern presented to the network, one output unit is declared winner. The winner backpropagates a signal that encodes the expected pattern template. If the current input pattern differs more than a predefined threshold from the backpropagated signal, the winner is temporarily disabled (by an orienting system) and the next closest unit is declared winner. The process continues until an output unit becomes a winner, considering the threshold. If no

one of the output units becomes a winner, a new output unit is defined to cluster the input pattern. Graphically, an ART1 network can be illustrated by Figure 2 (from [2]) where it appears with four input and six output 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 shown in Figure 3.

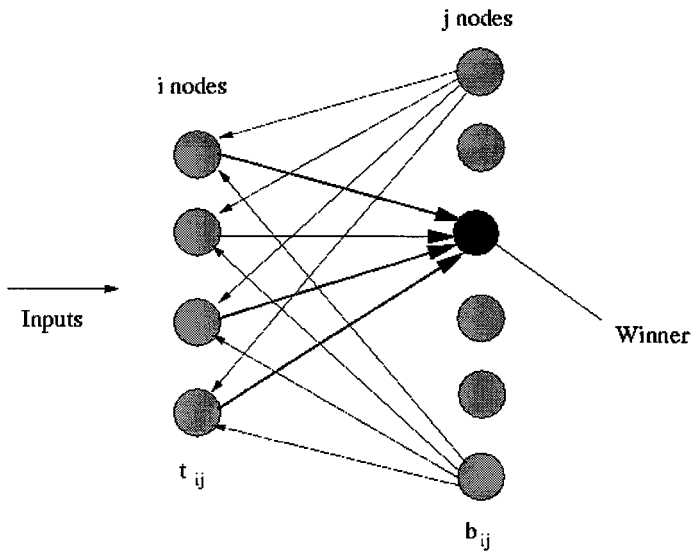


Figure 2. Architecture of an ART1 network

4.2 Combinatorial Neural Model

CNM is a hybrid architecture for intelligent systems that integrates symbolic and connectionist computational paradigms. This model is able to recognise 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 [4].

CNM uses supervised learning and a feedforward topology with one input layer, one hidden layer – here called combinatorial – and one output layer. Each neuron in the input layer corresponds to an evidence (denoted by e) of the world, expressed in an object-attribute-value form. In the combinatorial layer there are aggregative fuzzy AND neurons (denoted by c), each one connected to one or more neurons in the input layer by arcs with adjustable weights. The output layer contains one aggregative fuzzy OR neuron (also called hypothesis and denoted by h) for each possible class, linked to one or more neurons of the combinatorial layer. The synapses may be excitatory or inhibitory and are characterised by a strength value (weight) between zero - not connected – to one – fully connected synapses.

- **Step 1. 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.
- **Step 6. Disable best matching exemplar:** The output 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:**

$$t_{ij^*}(t+1) = t_{ij^*}(t)x_i$$

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

- **Step 8. Repeat by going to Step2:** first enable any node disabled in Step 6.

Figure 3. Training algorithm for ART1

The network is created completely uncommitted, 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 below:

PUNISHMENT_AND_REWARD_LEARNING_RULE

- Set the initial value of the accumulator in each arc of the network to 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 evidential flow (Reward)
 - Else backpropagate** from the hypothesis node until input nodes, decreasing the accumulator of each traversed arc by its evidential flow (Punishment).

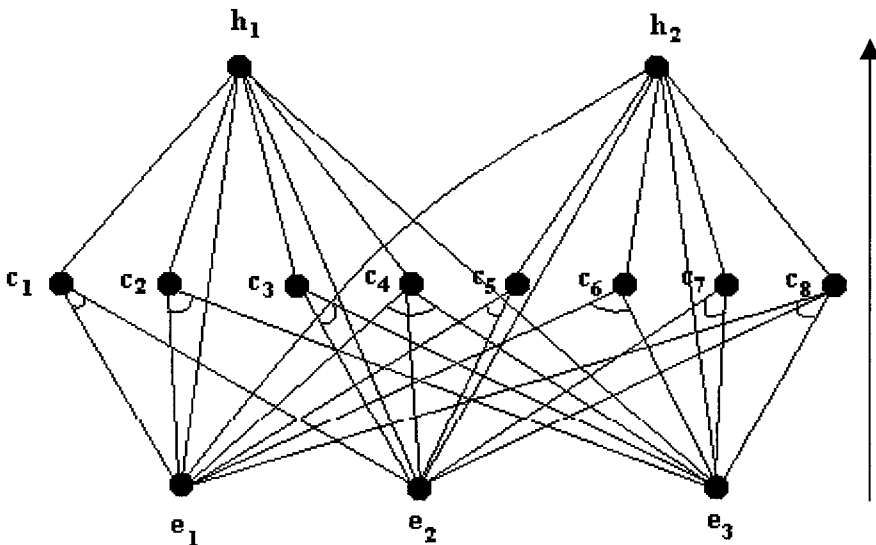


Figure 4. Combinatorial neural model

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 pruning of the network; and it is performed through 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 optimised and applied with success for data mining tasks ([1], [8] and [14]).

4.3 Functional model

The framework, instantiated with these two ANNs, is shown in the functional model in Figure 5. Layers F1 and F2 correspond to the ART1 network. F1 receives the codification of the input data in the framework, while F2 comprises the clusters identified in the process. Layers F2, F3, and F4 correspond to the CNM network.

Note that F2 layer represents the integration point between the two ANNs. During the whole process this layer is expanded with the creation of new clusters. Considering that the process occurs in just one data scan, it is important to observe what happens when a data point is linked to a specific cluster. This cluster (in F2) is regarded as a class and sent with the current data point to the classification part of the framework, which starts in F3.

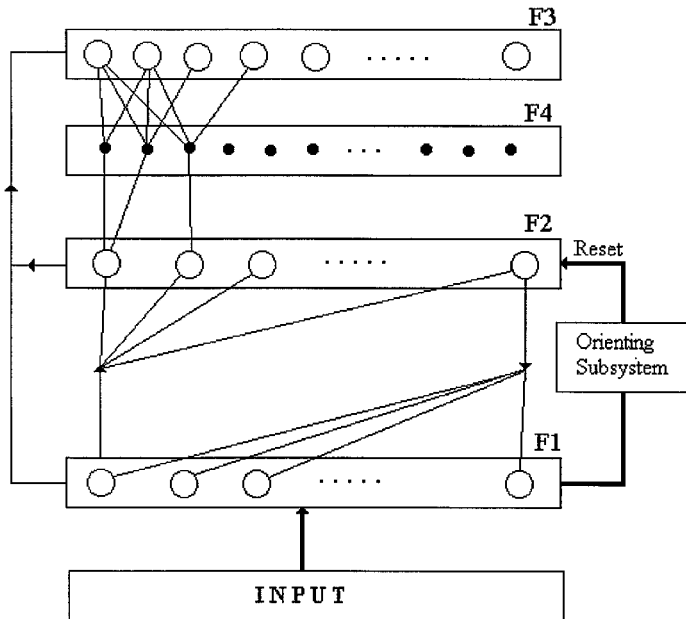


Figure 5. Framework instantiated with ART1 and CNM

5 Applying the framework to map wheat production

We validated the framework, using ART1 and CNM, through a case study in the agricultural research domain. Data related to 25 years of wheat plantation in 3015 localities of Brazil were collected including *municipality*, *geographical coordinates* (*altitude*, *longitude* and *latitude*), *amount cropped*, and *plantation area*. From these data, we considered the *amount cropped*, and *plantation area* fields, resulting in 50 fields for 25 years, to study homogeneous areas. As the two algorithms require discrete inputs, each variable was discretized into ten

classes, according to the experts practice. Figure 6 shows a snapshot of the main system window, containing the part of the intensional descriptions of a Class 4 in the RESULTS box. For each class it is listed the left side of each rule defining it. In the example, we could extract the following rule:

IF R1994 IS BETWEEN 576.2 AND 1072.4
 THEN CLASS 4 (10%, 66%, 281, 9.3%)

The values (10%, 66%, 281, 9.3%) express, respectively, the percentage of observations for this class, the confidence level for this rule, the support for this rule in number of observations and in percentage. The usual extensional descriptions are obtained directly from the data point labels.

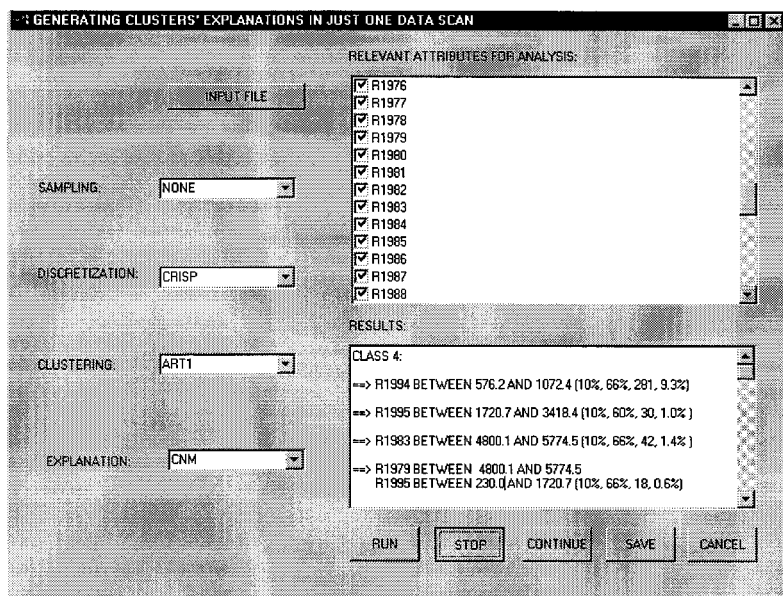


Figure 6. Results from the framework application

6 Contributions

The most important contribution of this research is a framework to help on the whole process of discovery from unlabeled data. Through this framework one can instantiate any two algorithms (for unsupervised and supervised learning) to identify groups composed by similar objects and describe their inner structure by the higher correlated features. Considering that each algorithm has its own merits, the possibility of applying different combinations of them and compare their results, seems to be particularly interesting. Other important contribution is the reuse of components such as the discretization and sampling processes.

References

- [1] 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.
- [2] Bigus, J. P., Data Mining with Neural Networks. [S.l.]: McGraw-Hill, 1996.
- [3] Bradley, P., Fayyad, U. and Reina, C. Scaling Clustering Algorithms to Large Databases. In: Proceedings of Fourth International Conference on Knowledge Discovery and Data mining. New York, NY: AIII Press, 1998.
- [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, pp.233-287, 1988.
- [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] Farnstrom, F., Lewis, J., Elkan, C. Scalability for Clustering Algorithms Revisited, SIGKDD Explorations, v. 2, Issue 1, pp.51-57, 2000.
- [8] Feldens, M.A., Castilho, J.M.V., Data mining with the Combinatorial Rule Model: an application in a health-care relational database. IN: CLEI, 23.,1997, Valparaizo, Chile, **Proceedings ...** Valparaízo: CLEI, 1997. V.1, pp.135-145.
- [9] Freeman, J. A., Skapura, D. M., Neural Networks, Algorithms, Applications, and Program Techniques. [S.l.]: Addison-Wesley, pp.293-339, 1992.
- [10] Langley, P. The Computer-Aided Discovery of Scientific Knowledge. In: Proceedings of the First International Conference on Discovery Science, Fukuoka, Japan, 1998.
- [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, pp.831-847, 1998.
- [13] Murphy, G, and Medin, D., The Role of Theories in Conceptual Coherence. Psychological Review, 92(3):289-316, 1985.
- [14] 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.