

Combining Argumentation and Clustering Techniques in Pattern Classification Problems

Sergio Alejandro Gómez¹

Carlos Iván Chesñevar²

¹Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (Miembro del IICyTI Instituto de Investigación en Ciencia y Tecnología Informática). — Depto. de Cs. e Ing. de la Computación Universidad Nacional del Sur — Av. Alem 1253 – B8000CPB Bahía Blanca – REPÚBLICA ARGENTINA
TEL/FAX: (+54) (291) 459 5135/5136 – EMAIL: sag@cs.uns.edu.ar

²Grupo de Investigación en Inteligencia Artificial — Departament d'Informàtica Universitat de Lleida – Campus Cappont – C/Jaume II, 69 – E-25001 Lleida, SPAIN
TEL/FAX: (+34) (973) 70 2764 / 2702 – EMAIL: cic@eup.udl.es

KEYWORDS: Machine Learning, Defeasible Argumentation, Neural networks, Fuzzy Adaptive Resonance Theory.

Abstract

Clustering techniques can be used as a basis for classification systems in which clusters can be classified into two categories: *positive* and *negative*. Given a new instance e_{new} , the classification algorithm is applied to determine to which cluster c_i it belongs and the label of the cluster is checked. In such a setting clusters can overlap, and a new instance (or example) can be assigned to more than one cluster. In many cases, determining to which cluster this new instance actually belongs requires a qualitative analysis rather than a numerical one.

In this paper we present a novel approach to solve this problem by combining defeasible argumentation and a clustering algorithm based on the Fuzzy Adaptive Resonance Theory neural network model. The proposed approach takes as input a clustering algorithm and a background theory. Given a previously unseen instance e_{new} , it will be classified using the clustering algorithm. If a conflicting situation arises, argumentation will be used in order to consider the user's preference criteria for classifying examples.

1 Introduction

Clustering techniques can be used as a basis for classification systems in which clusters can be classified into two categories: *positive* and *negative*. Most clustering techniques consider first a database of *training examples* $\{e_1, e_2, \dots, e_n\}$ as an input for a classification algorithm, determining a number of clusters c_1, \dots, c_m which are labelled as either positive or negative. Given a new instance e_{new} , the classification algorithm is applied to determine to which cluster c_i it belongs and the label of the cluster is checked. Should the cluster c_i be labelled as positive (negative), then the instance e_{new} is regarded as positive (negative). This approach has been exploited in the web document filtering agent Querando! [GL01a, GL01b] and in the counter-propagation neural network model [FS93, Ska96, RR95, Was89]. It must be remarked that in such classification algorithms we adopt the usual approach in machine learning [Mit97] by distinguishing between training examples (training data that has a label provided by the user) and instances (data that does not have such a label).

Given a collection of training examples describing some concept in function of time, a clustering algorithm can be used to classify these examples into clusters (defined in terms of some similarity metrics). Each one of these clusters is assigned a label provided by the user (either *positive* or *negative*). The label *positive* indicates that the examples in the cluster belong to the concept whereas the label *negative* indicates that the examples in the cluster do not belong

to the concept. Since clusters can overlap, a new instance (or example) can be assigned to more than one cluster. When such clusters containing the instance are contradictory (*i.e.*, some of them are labelled as positive, whereas others are negative), an undesirable situation occurs. So the problem to be solved consists of determining to which clusters this new instance actually belongs.

In this article we show a novel approach to solve the above problem by combining defeasible argumentation and a clustering algorithm based on the Fuzzy Adaptive Resonance Theory neural network model. The proposed approach takes as input a clustering algorithm and an underlying background theory T , formalized in terms of *defeasible logic programming* [GS02]. Given a previously unseen instance or training example, it is classified using the clustering algorithm, modifying the cluster structure accordingly. If a conflicting situation arises, the background theory T will be used in order to consider the user's preference criteria for classifying examples. The ultimate outcome involves determining whether the new instance is positive, negative or unknown.

The article is structured as follows. First in Section 2 the fundamentals of the theory of defeasible argumentation are introduced. Then in Section 3 we present a particular learning algorithm for neural networks, called Fuzzy Adaptive Resonance Theory (ART). In Section 4 we show how to model clustering in terms of Fuzzy ART learning and defeasible argumentation, characterizing the users preference criteria in terms of a *defeasible logic program*. Section 5 summarizes previous work related to combining argumentation with other machine learning techniques. Finally Section 6 discusses the main conclusions that have been obtained and outlines some future research work.

2 Defeasible Argumentation

Artificial Intelligence has long dealt with the issue of finding a suitable formalization for commonsense reasoning. Defeasible argumentation [SL92, CML00, PV99] has proven to be a successful approach in many respects, since it naturally resembles many aspects of human commonsense reasoning. The growing success of argumentation-based approaches has caused a rich crossbreeding with other disciplines, providing interesting results in different areas such as legal reasoning, medical diagnosis, and decision support systems. As pointed out in [CRL00], most argument-based frameworks share a number of common notions, namely:

1. **Knowledge Base. Underlying logical language:** Most argument-based frameworks involve a knowledge base $K = (\Pi, \Delta)$ which provides *background knowledge* for an intelligent agent formalized in a first-order logical language L . This background knowledge typically involves a set Π of *strict rules* and *facts* as well as a set Δ of *defeasible rules*.
2. **Argument:** An *argument* is a defeasible proof obtained from the knowledge base K by applying suitable (defeasible) inference rules associated with the underlying logical language L .
3. **Dialectical reasoning:** Given two arguments A and B , conflict (or attack) among arguments arises whenever A and B cannot be simultaneously accepted (typically because of some kind of logical contradiction). Many argument systems provide a preference criterion which defines a partial order among arguments, allowing to determine whether A should be preferred over B . This defines a *defeat* relationship. Given the set $Args$ of arguments obtained from a knowledge base K , it holds that $attacks \subseteq Args \times Args$,

and *defeats* \subseteq *attacks*. In order to determine whether a given argument A is ultimately undefeated (or *warranted*), a dialectical process is recursively carried out, where defeaters for A , defeaters for these defeaters, and so on, are taken into account.

Defeasible logic programming (DeLP) is a particular formalization of defeasible argumentation based on logic programming [GS02]. A defeasible logic program $Prog$ is a set $K = (\Pi, \Delta)$ of Horn-like clauses, in which defeasible and strict rules can be distinguished. Deriving literals in DeLP results in the construction of arguments. Since arguments may be in conflict, a dialectical analysis is needed. Given a DeLP program $Prog$ and a query q , the final answer to q wrt $Prog$ takes such dialectical analysis into account. An in-depth treatment of DeLP is outside the scope of this paper. For details we refer the reader to [GS02].

Argumentation provides mostly a non-numerical, *qualitative* setting for commonsense reasoning, which does not seem adequate for modelling the pattern classification problem described in the previous section. Clearly, pattern classification relies on *quantitative aspects* of the data involved (such as numeric attributes or probabilities). As we will see in the next sections, our final goal is to develop a hybrid approach in which both quantitative and qualitative features required for pattern classification are combined. Qualitative aspects will be captured in terms of defeasible argumentation using DeLP, whereas quantitative ones will be captured by using the so-called *Fuzzy Adaptive Resonance Theory*, described in the next section.

3 Fuzzy Adaptive Resonance Learning

Adaptive Resonance Theory (ART) [FS93, RR95] is a class of neurally inspired models of how the brain performs clustering and classification of sensory data, and associations between the data and representation of concepts. Fuzzy ART performs unsupervised learning of categories under continuous presentation of inputs, through a process of ‘adaptive resonance’ in which the learned patterns adapt only to relevant inputs, but remain stable under irrelevant or insignificant ones. Thus the ART models solve the so-called stability-plasticity dilemma where new patterns are learned without forgetting those learned previously.

The Fuzzy Adaptive Resonance Theory neural network model is a kind of ART neural network that accepts analog inputs (in the real interval $[0, 1]$) [LCS99, LC99]. Familiar inputs activate the category, whereas unfamiliar inputs trigger either adaptive learning by an existing category or a commitment of a new category. The behaviour of Fuzzy ARTs lends itself well to simple geometrical interpretation of category prototypes as *hyperrectangles* in the input space. These rectangles are allowed to overlap each other. Although Fuzzy ART always responds the same way to a familiar input —it recalls the smallest hyperrectangle containing this input—, the overlaps are inconvenient if categories are mutually exclusive.

Next we present a synthesized version of the Fuzzy ART algorithm as well as the geometrical interpretation of its fast learning rule. For further details we refer the reader to [LCS99].

3.1 Learning Algorithm

The Fuzzy ART learns a *categorization* or *clustering* of a sequence of examples presented to the network. Its learning algorithm is as follows [LCS99]:

Category initialization: Each category j is represented by an $2M$ -dimensional vector $w_j = (w_{j1}, \dots, w_{j2M})$ of adaptive weights. Before any input presentation occurs, each category is initially uncommitted, and its weights are initialized to one.

Complement coding: To avoid a category proliferation problem, the input is normalized by complement coding. Let a be an M -dimensional vector (a_1, \dots, a_M) , where $0 \leq a_i \leq 1$. The complement coded input I is obtained as:

$$I = (a_1, \dots, a_M, 1 - a_1, \dots, 1 - a_M) = (a, a^c) \quad (1)$$

Category choice: Upon presentation of an input I , a *choice function* T_j is computed for each category j :

$$T_j = \frac{|I \wedge w_j|}{\alpha + |w_j|} \quad (2)$$

The norm operator $|\cdot|$ is defined as $|x| = \sum_{i=1}^{2M} |x_i|$, the symbol \wedge denotes the fuzzy AND operator, i.e. $x \wedge y = (\min(x_1, y_1), \dots, \min(x_{2M}, y_{2M}))$, and α is a user-defined parameter, $\alpha > 0$. The category J for which the choice function T_J is maximal is chosen for the vigilance test.

Vigilance test: The similarity between w_J and I is compared to a parameter ρ called *vigilance*, $0 \leq \rho \leq 1$, in the following test:

$$\frac{|I \wedge w_J|}{|I|} \leq \rho. \quad (3)$$

If the test is passed, then resonance occurs and learning takes place. If the test is failed, then mismatch reset occurs: the value of T_j is set to -1 for the duration of the current input presentation, another category is chosen and the vigilance test is repeated. Categories are searched until one that meets Eq. 3 is found. This category is said to be selected for I . It is either already committed or uncommitted, in which case it becomes committed during resonance.

Resonance: During resonance, the weight vector w_J of the selected category is updated according to:

$$w_J(t+1) = \beta(I \wedge w_J(t)) + (1 - \beta)w_J(t) \quad (4)$$

where β is a learning rate parameter, $0 < \beta \leq 1$. When $\beta = 1$, this special case is called fast learning. Once resonance is finished, a new input may be presented and the last three steps repeated.

3.2 Geometrical Interpretation of Learning

The Fuzzy ART has a very well known geometrical interpretation [LCS99]. Each weight vector w_j may be written in the form $w_j = (u_j, v_j^c)$ where u_j and v_j are M -dimensional vectors corresponding to the two opposite corners of a hyperrectangle R_j . With fast learning Eq. 4 reduces to $w_J(t+1) = I \wedge w_J(t)$ and the corners of R_j are updated by $u_J(t+1) = a \wedge u_J(t)$ and $v_J(t+1) = a \vee v_J(t)$, where \vee denotes the fuzzy OR operator, that is, $x \vee y = (\max(x_1, y_1), \dots, \max(x_M, y_M))$. When a committed category is selected, R_J expands to the minimum hyperrectangle containing both R_J and the input a . If a lies inside of R_J , then R_J is unchanged. Thus when a category j is committed, its size can only grow or remain the same.

3.3 Fuzzy ART as a Basis for Supervised Learning

As explained above, the Fuzzy ART neural network learns a clustering of the input space. If we choose to label each one of these clusters either as positive or negative depending on a label assigned to training examples, the Fuzzy ART can be used as a basis for supervised learning.

Given one particular cluster c , it should be noted that both positive *and* negative examples could be present in c , leading to an inconsistent situation. A preference criterion is required to solve such a situation. Some possible preference criteria could be as follows: (1) a cluster is labelled as positive if the count of positive training examples is greater than the count of negative training examples; (2) the cluster label is the label of the example that triggered the cluster creation, or, (3) the cluster label is the label of the last example belonging to the cluster.

4 Modelling Clustering with Fuzzy ART Learning and DeLP

In this section we will describe the main aspects concerning how to model clustering in terms of Fuzzy ART learning and defeasible argumentation. First, in section 4.1 we will consider how to represent user-provided examples as a number of *facts* of a defeasible logic program. Then in section 4.2 we analyze how to solve the conflict among conflicting clusters by incorporating preference criteria modelled in terms of strict and defeasible *rules*. Finally, in section 4.3 we show how the inference procedure of DeLP allows us to determine whether a particular cluster labelling is ultimately to be preferred by performing a dialectical analysis.

4.1 Building a Knowledge Base with Training Examples

As discussed in the introduction, the usual methodology when performing clustering in our context is as follows: the user will provide a number of *points* separated in two classes: (1) training examples (labelled as positive or negative) or (2) instances (unlabelled points). As new labelled points are provided by the user, the underlying neural network is fed with this information, modifying the cluster structure learnt so far. At the same time, a DeLP program consisting of *facts* based on this information is generated.

Example 1 Suppose that the user provides as an input some particular points p_1, p_2, p_3, p_4 , and p_5 labelled as positive, negative, positive, negative, and positive, resp., as well as some other sample points. Finally, an instance p_n is provided at time t_n . This would result in a cluster structure learnt by the neural network based on the Fuzzy ART approach, which would remain hidden for the user. At the same time, a DeLP program *Prog* as the one shown below will be generated:

$example(p_1, positive, t_1) \leftarrow$	$triggeredCreation(p_1, c_1, t_1) \leftarrow$
$example(p_2, negative, t_2) \leftarrow$	$t_1 < t_2 \leftarrow$
$triggeredCreation(p_2, c_2, t_2) \leftarrow$	$example(p_3, positive, t_3) \leftarrow$
$t_1 < t_3 \leftarrow$	$t_2 < t_3 \leftarrow$
$triggeredCreation(p_3, c_3, t_3) \leftarrow$	$example(p_4, negative, t_4) \leftarrow$
$triggeredCreation(p_4, c_4, t_4) \leftarrow$	$t_1 < t_4 \leftarrow$
$t_2 < t_4 \leftarrow$	$t_3 < t_4 \leftarrow$
$example(p_5, positive, t_5) \leftarrow$	$triggeredCreation(p_5, c_5, t_5) \leftarrow$
$t_1 < t_5 \leftarrow \dots$	$t_4 < t_5 \leftarrow$
\dots	$instance(p_n, t_n) \leftarrow$
$t_1 < t_n \leftarrow \dots$	$t_{n-1} < t_n \leftarrow$

Note that every new example corresponding to a point P labelled as L at time T causes a new fact $example(P, L, T)$ to be added to the program $Prog$. When the dynamics of the neural network determines that a new cluster is to be created, a new fact $triggeredCreation(P, C, T)$ is added to the program $Prog$. This fact stands for “a new cluster C was created at time T because of occurrence of point P ”. Note that the time sequence in which points are provided by the user is automatically traced by adding literals of the form $t_i < t_j$.¹ Finally, the predicate $instance(P, T)$ denotes that a new (unlabelled) instance point P was presented to the system at time T . The cluster associated with this point P is still undetermined.

4.2 Modelling Attack and Defeat in Fuzzy ART Classification

In the Fuzzy ART setting, conflict appears when an instance never seen before is classified by *more than one cluster with different labels*. Suppose for instance that a new pattern x is presented to the Fuzzy ART neural network and it determines that x belongs to *both* cluster i and j , and cluster i is labelled as positive while cluster j is labelled as negative. In the literature[LCS99, LC99], the above situation is solved nondeterministically by making a random choice.

We think that the above situation can be modelled as a special kind of “*attack* between clusters”. In defeasible argumentation, the notion of attack is defined as a conflict (usually captured in terms of some kind of logical contradiction) between two arguments. In the Fuzzy ART setting, given a point belonging to several overlapping clusters, conflict appears whenever two (or more) overlapping clusters are labelled differently. Following the same line of reasoning, the notion of attack in defeasible argumentation is refined as a defeat relationship [SL92] by adding a preference criterion among arguments (e.g. specificity). When deciding among overlapping clusters, several preference criteria can be defined to solve conflicts in the Fuzzy ART setting, namely:

- The cluster with newer information wins. In this case we use a *novelty* approach: newer examples are used to solve the conflict as they are considered more accurate than previously seen examples. In this way both incremental learning and changes in the user interest can be better modeled.
- The cluster that subsumes more examples wins. In this case we use a kind of *support* criterion: the more examples are subsumed by a cluster prototype, the more veritable the cluster label is.
- The smallest cluster wins. In this case we use an specificity-based approach: the smaller a cluster is, the more specific it is assumed to be.

It must be noted that the above criteria could be provided by the user in terms of rules corresponding to a defeasible logic program. The following example extends Example 1 by providing some rules for modelling preference between conflicting clusters.

Example 2 Consider the strict and defeasible rules presented in Figure 1. Such rules model preference criteria among conflicting clusters. Next we will explain briefly the meaning of such rules and their associated predicates. Predicate *activatesNonDeterm* stands for “point

¹We use this notation for clarity purposes instead of the usual prefix notation *lessThan*(t_i, t_j). Note that only facts relating the time stamps of the last two points p_n and p_{n+1} are required, as other relationships between time stamps can be easily derived by transitivity from those facts.

$$\begin{aligned}
\textit{opposite}(\textit{positive}, \textit{negative}) &\leftarrow & (5) \\
\textit{opposite}(X, Y) &\leftarrow \textit{opposite}(Y, X). & (6) \\
\sim \textit{is}(L_1, P, T) &\leftarrow \textit{is}(L_2, P, T), \textit{opposite}(L_1, L_2). & (7) \\
\textit{labelledAs}(L, C, T) &\multimap \textit{example}(P, L, T_0), & \\
&\textit{triggeredCreation}(P, C, T_0), T_0 < T. & (8) \\
\textit{is}(L, P, T) &\multimap \textit{belongsTo}(P, C, T), \textit{labelledAs}(L, C, T). & (9) \\
\sim \textit{labelledAs}(L_1, C_1, T_1) &\multimap \textit{intersects}(C_1, C_2, T), \textit{labelledAs}(L_2, C_2, T_2), & \\
&T_1 < T_2, \textit{opposite}(L_1, L_2). & (10) \\
\sim \textit{is}(L_2, P, T) &\multimap \textit{isInstance}(P, T), \textit{belongsTo}(P, C_1, T), & \\
&\textit{labelledAs}(L, C_1, T), \textit{belongsTo}(P, C_2, T), & \\
&\textit{labelledAs}(L_2, C_2, T), \textit{opposite}(L_1, L_2), & \\
&N = \textit{count}(C_1, T), M = \textit{count}(C_2, T), & \\
&N \geq M. & (11) \\
\textit{activatesNonDeterm}(C, P, T) &\leftarrow \text{Fuzzy ART inner workings. . .} & (12) \\
\textit{intersects}(C_1, C_2, T) &\leftarrow \text{Fuzzy ART inner workings. . .} & (13) \\
\textit{belongsTo}(P, C, T) &\multimap \textit{activatesNonDeterm}(C, P, T) & (14)
\end{aligned}$$

Figure 1: *Modelling preference criteria between overlapping clusters in DeLP.*

P activates nondeterministically cluster unit C at time T ". Rules 5 and 6 say that *positive* and *negative* labels for a cluster oppose each other, representing opposite concepts. Rule 7 says that a cluster is not positive at time T if it is negative at time T and viceversa. Rule 8 says that a label L of a cluster C at time T agrees with the label assigned by the user to point P when such point triggered the creation of a such a cluster at time T_0 . Rule 9 says that the label L of a point P is the label of the cluster C containing it. Rule 10 says that when a cluster intersects another cluster with opposite label and the latter was labelled later than the former, then the newer cluster wins for deciding the resulting labelling. Rule 11 says that when a point P belongs to two clusters C_1 and C_2 with opposing labels L_1 and L_2 , the cluster with subsumes more examples is the winner. Rules 12 and 13 model the inner behavior of the Fuzzy ART neural networks. Rule 12 denotes the nondeterministic activation of cluster unit C by point P at time T . Rule 13 denotes that clusters C_1 and C_2 have non empty intersection at time T . Finally, rule 14 stands for the fact that a point P typically belongs to cluster C at time T if it activates non deterministically cluster unit C at time T .

Usually, the Fuzzy ART approach to clustering for classifying a previously unseen instance (represented in our case as a new point p_n) is as follows. The point p_n is presented to the Fuzzy ART neural net. The network gives as an answer a cluster unit that codifies the point (modifying the cluster structure computed so far accordingly). After that the cluster unit label is determined. As stated before, when clusters are overlapping, a point may belong to several clusters simultaneously. If the cluster labels are distinct, an undesirable situation occurs because the answer has to be chosen nondeterministically.

A DeLP program as the one detailed in the previous examples can provide additional, qualitative information for making such a decision. For the sake of example, suppose that the

cluster structure built so far is that of Figure 2: there are five overlapping clusters, and there is a new instance p_n lying in the intersection of clusters c_1 , c_2 , and c_4 that needs to be classified.

Instead of choosing the answer randomly², *arguments* can be built that support what decision is to be made. Next we show two arguments A_1 and A_2 . The first supports the conclusion that the point p_n is positive, as it belongs to cluster c_1 and cluster c_1 is labelled as positive. The second supports the opposite claim, point p_n is negative, since it belongs to cluster c_2 and cluster c_2 is labelled as negative. Formally:

$$A_1 : \langle \{ is(positive, p_n, t_n) \rightarrow belongs(p_n, c_1, t_n), labelledAs(positive, c_1, t_n). \}, is(positive, p_n, t_n) \rangle$$

$$A_2 : \langle \{ \sim is(positive, p_n, t_n) \rightarrow isInstance(p_n, t_n), belongs(p_n, c_2, t_n), labelledAs(negative, c_2, t_n). \}, is(negative, p_n, t_n) \rangle$$

Both arguments were derived from the information provided by the program *Prog* given in the previous examples. Both arguments cannot be accepted, as they lead to conflicting conclusions. Next we will outline how such a conflict is solved.

4.3 Deciding Clustering Labels in Terms of a Dialectical Analysis

Defeasible argumentation frameworks allow to perform a dialectical analysis for arguments in conflict. Given an argument A for a hypothesis h , all possible *counterarguments* B_1, \dots, B_k that defeat A are considered. This analysis is performed recursively on those counterarguments, by considering all counter-counterarguments that defeat every B_i , and so on. This results in a tree-like structure T rooted in $\langle A, h \rangle$, where every node (argument) can be labelled as D (defeated) or U (undefeated). A given argument B in T is a U -node iff it has no defeaters, or every children of B is a D -node. A given argument B in T is a D -node iff it has at least one children node labelled as U -node. If the root of the tree T is labelled as a U -node, such an argument is said to be *warranted*, i.e. ultimately accepted. It must be remarked that DeLP performs such a dialectical analysis automatically when answering a given query.³

In our case, the above analysis is started in DeLP by analyzing whether there exists a warranted argument supporting that an new unlabelled instance p_n has to be labelled as positive. If this is not the case, the same analysis is performed, by considering the label “negative”.

Example 3 Suppose that the cluster structure built so far by the Fuzzy ART is the one of Figure 2, and a new point p_n is to be classified, assuming a DeLP program as the one described before as a background knowledge. As clusters overlap when analyzing the labelling for p_n , the DeLP inference engine will try to find a warranted argument supporting the fact that p_n is to be labelled as positive. The argument A_1 previously shown is found:

$$A_1 : \langle \{ is(positive, p_n, t) \rightarrow belongs(p_n, c_1, t_n), labelledAs(positive, c_1, t_n). \}, is(positive, p_n, t_n) \rangle$$

DeLP will start the search for defeaters in a depth-first fashion. An argument A_2 which attacks and defeats A_1 will be found, namely:

$$A_2 : \langle \{ \sim is(positive, p_n, t_n) \rightarrow isInstance(p_n, t_n), belongs(p_n, c_2, t_n), labelledAs(negative, c_2, t_n). \}, is(negative, p_n, t_n) \rangle$$

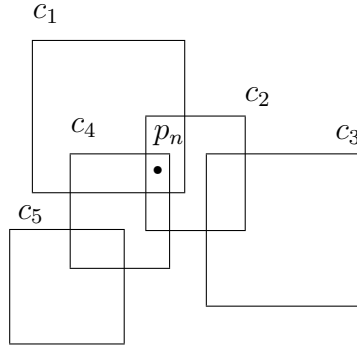


Figure 2: *Nondeterministic classification of point p_n by cluster units c_1 , c_2 , and c_4 .*

Nevertheless, a new argument A_3 can be used in turn to defeat argument A_2 : “Cluster c_2 should not be labelled as negative because clusters c_2 and c_3 have examples in common and cluster c_3 is labelled as positive and cluster c_3 is composed of newer examples.” Formally:

$$A_3 : \langle \{ \sim \text{labelledAs}(\text{negative}, c_2, t_n) \rightarrow \text{intersects}(c_2, c_3, t_n), \text{labelledAs}(\text{positive}, c_2, t_n) \}, \\ \sim \text{labelledAs}(\text{negative}, c_2, t_n) \rangle$$

In this point, DeLP will find a new argument A_4 that defeats argument A_1 : “The point p_n is not positive because it belongs to cluster c_4 , cluster c_4 is labelled as negative, and cluster c_4 subsumes more examples than c_1 does.” Formally:

$$A_4 : \langle \{ \sim \text{is}(\text{positive}, p_n, t_n) \rightarrow \text{isInstance}(p_n, t_n), \text{belongsTo}(p_n, c_4, t_n), \\ \text{labelledAs}(\text{negative}, c_4, t_n), \text{belongsTo}(p_n, c_1, t_n), \text{labelledAs}(\text{positive}, c_1, t_n), \\ \text{opposite}(\text{negative}, \text{positive}), n_4 = \text{count}(c_4, t_n), \\ n_1 = \text{count}(c_1, t_n), n_4 \geq n_1 \}, \sim \text{is}(\text{positive}, p_n, t_n) \rangle$$

But argument A_4 will be defeated by another argument A_5 —having the same spirit of argument A_3 —regarding the non-null intersection of the positively labelled cluster c_5 with the negatively labelled cluster c_4 :

$$A_5 : \langle \{ \sim \text{labelledAs}(\text{negative}, c_4, t_n) \rightarrow \text{intersects}(c_4, c_5, t_n), \text{labelledAs}(\text{positive}, c_5, t_n) \}, \\ \sim \text{labelledAs}(\text{negative}, c_4, t_n) \rangle$$

The above analysis results in the dialectical tree depicted in Figure 3, where nodes are arguments, arrows stand for a defeat relationship, and markings D and U stand for *defeated* and *undefeated*, resp. The root node turns out to be a U -node, and hence it is warranted.

4.4 Combining DeLP and Fuzzy ART: Discussion

Figure 4 shows a sketch of an algorithm that combines the use of DeLP and the Fuzzy ART for determining the classification of an instance P presented at time T wrt a concept represented as a set of labeled clusters $\{c_1, \dots, c_m\}$. The algorithm takes as input a Fuzzy ART neural network, a background theory $Prog$ in DeLP (characterizing a set of examples and preference

²Under certain special conditions, the Fuzzy ART will answer with the smallest hyperrectangle containing its input [LCS99].

³See [GS02] for details.

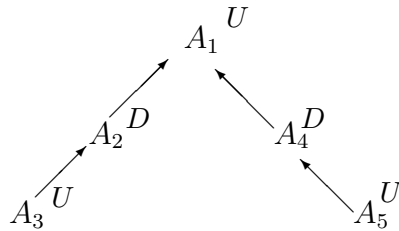


Figure 3: *Dialectical analysis yielding the conclusion that point p_n is a positive instance.*

criteria), and the data corresponding to a new unlabelled instance P . Such an instance P is first classified using the Fuzzy ART neural network (modifying the cluster structure accordingly if needed). In case such a classification cannot be successfully performed by the network, then the underlying program $Prog$ is used for determining whether the instance is either positive or negative or unknown according to: (1) *positive* if an argument $\langle A_{c_i}, is(positive, P, T) \rangle$ is warranted from $Prog$; (2) *negative* wrt $\langle A_{c_i}, is(negative, P, T) \rangle$, and, (3) *unknown* if neither (1) nor (2) can be proven.

It must be noted that there exists a theorem [GS02] that ensures that if some argument $\langle A, h \rangle$ is warranted, then it does not exist a warranted argument for the opposite conclusion, i.e., $\langle B, \sim h \rangle$. As a consequence, when analyzing the label associated with a new instance P , it cannot be the case that the underlying background theory allows us to conclude that P is warranted to be positive and negative simultaneously.

5 Related Work

The combination of machine learning and argumentation is a recent development. In a recent paper [GC03], we explored the combination of machine learning techniques and argumentation systems. To the best of our knowledge, there have been no similar approaches in this direction.

There are many texts that explore the field of neural network applications [FS93, LCS99, RR95, Ska96, Was89]. The area of clustering algorithms have a wide range of applications which include image processing [MGA94], information retrieval [Ras92] and text filtering [Hon97, GL01a, GL01b, MMLP97], and robot control [ZSS97]. Although all of these works solve effectively the problem of clustering, none of them uses argumentation as described in this paper. In particular, the pitfalls of Fuzzy ART are exploited as an advantage for doing multiple categorization in [LCS99], proposing a variation on the Fuzzy ART model.

Shavlik and Towel use a rule set to initialize a backpropagation neural network training algorithm [Mit97, pp. 340–363]. In that case, rules are used for initializing the neural network weights, while we use defeasible rules for revising a neural network classification a posteriori. In [JG03] an algorithm is proposed for inducing a defeasible theory from a set of training examples. In [IK97], the authors develop a method to generate non-monotonic rules with exceptions from positive/negative examples and background knowledge in Inductive Logic Programming. While these works induce a defeasible theory from examples, the proposed approach uses a defeasible theory for improving an incremental categorization. Other hybrid approaches include the an agent collaboration protocol for database initialization of a memory based reasoning algorithm in the context of an e-mail filtering system [LMM97]. This work also uses rules for improving learning speed. In contrast, the proposal presented in this paper is aimed to improve

ALGORITHM ClassifyNewInstance
INPUT: Fuzzy ART Net F , DeLP $Prog$, Point P , Timestamp T
OUTPUT: positive, negative, unknown { Classification of point P . }
BEGIN
 Propagate Point P with timestamp T through Net F
 $WinnerUnitSet := GetWinnerUnitSet(F)$
 { $WinnerUnitSet$ is the set of clusters that contain point P . }
 IF All unit labels in $WinnerUnitSet$ are positive **OR**
 All unit labels in $WinnerUnitSet$ are negative **THEN**
 RETURN Label of any unit in $WinnerUnitSet$
 ELSE
 Perform dialectical analysis to determine if there exists an argument A
 such that $\langle A, is(positive, P, T) \rangle$ is warranted.
 IF $\langle A, is(positive, P, T) \rangle$ is warranted **THEN RETURN** Label=positive
 ELSE
 Perform dialectical analysis to determine if there exists an argument A
 such that $\langle A, is(negative, P, T) \rangle$ is warranted.
 IF $\langle A, is(negative, P, T) \rangle$ is warranted
 THEN RETURN Label=negative
 ELSE RETURN Label=unknown
 END
 END
END

Figure 4: An algorithm for relating an argument-based framework with the Fuzzy ART model.

learning precision.

6 Conclusions and Future Work

The growing success of argumentation-based approaches has caused a rich cross-breeding with other disciplines, providing interesting results in different areas such as legal reasoning, medical diagnosis and decision support systems.

We contend that existing frameworks for defeasible argumentation can be enriched by integrating them with machine learning techniques, as the one discussed in this paper, making them more attractive and suitable for other several application areas. Following this research line, in this paper we have introduced an approach that integrates a machine learning technique (such as the Fuzzy ART neural networks) with an argumentation framework.

Part of our current research work involves the development of a more refined version of the algorithm presented in this paper in order to test it with respect to some benchmark standard collections⁴.

References

- [CML00] CHESÑEVAR, C. I., MAGUITMAN, A., AND LOUI, R. Logical Models of Argument. *ACM Computing Surveys* 32, 4 (December 2000), 337–383.

⁴For instance <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

- [CRL00] CARBOGIM, D., ROBERTSON, D., AND LEE, J. Argument-based applications to knowledge engineering. *The Knowledge Engineering Review* (2000).
- [FS93] FREEMAN, J., AND SKAPURA, D. *Redes Neuronales. Algoritmos, aplicaciones y técnicas de programación*. Addison-Wesley/Díaz de Santos, 1993.
- [GC03] GÓMEZ, S. A., AND CHESÑEVAR, C. I. Integrating Defeasible Argumentation and Machine Learning Techniques: A Preliminary Report. *V Workshop de Investigadores en Ciencias de la Computación (WICC 2003)* (2003), 320–324.
- [GL01a] GÓMEZ, S. A., AND LANZARINI, L. Aplicación de redes neuronales al filtrado de documentos. *III Workshop de Investigadores en Ciencias de la Computación (WICC 2001)* (2001), 131–133.
- [GL01b] GÓMEZ, S. A., AND LANZARINI, L. Querando!: Un agente de filtrado de documentos web. *VII Congreso Argentino de Ciencias de la Computación (CACIC 2001)* (2001), 1205–1217.
- [GS02] GARCÍA, A. J., AND SIMARI, G. R. Defeasible Logic Programming an Argumentative Approach. *Theory and Practice of Logic Programming (To appear)* (2002). Available at www.cs.uns.edu.ar/~grs.
- [Hon97] HONKELA, T. *Self-organizing maps in Natural Language Processing*. PhD thesis, Helsinki University of Technology, 1997.
- [IK97] INOUE, K., AND KUDOH, Y. Learning Extended Logic Programs. In *Proc. of the 15th Int. Conf. on Artificial Intelligence* (1997), Morgan Kaufmann.
- [JG03] JOHNSTON, B., AND GOVERNATORI, G. An algorithm for the induction of defeasible logic theories from databases. *Australian Computer Society, Inc. In Fourteenth Australasian Database Conference (ADC2003), Adelaide, Australia. Conferences in Research and Practice in Information Technology, Vol. 17. Xiaofang Zhou and Klaus-Dieter Schewe, Ed.* (2003).
- [LC99] LUBKIN, J., AND CAUWENBERGHS, G. VLSI Implementation of Fuzzy Adaptive Resonance and Learning Vector Quantization. *Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio- Inspired Systems. Granada, Spain.* (1999), 147.
- [LCS99] LAVOIE, P., CRESPO, J., AND SAVARIA, Y. Generalization, discrimination, and multiple categorization using adaptive resonance theory. *IEEE Transactions on Neural Networks Vol. 10, No. 4* (July 1999).
- [LMM97] LASHKARI, Y., METRAL, M., AND MAES, P. Collaborative interface agents. In *Readings in Agents*. Morgan Kaufmann, 1997, pp. 111–116.
- [MGA94] MARAVALL GÓMEZ ALLENDE, D. *Reconocimiento de Formas y Visión Artificial*. Ed. Addison-Wesley Iberoamericana, 1994.
- [Mit97] MITCHELL, T. *Machine Learning*. Mc Graw Hill, 1997.
- [MMLP97] MOSTAFA, J., MUKHOPADHYAY, S., LAM, W., AND PALAKAL, M. A multilevel approach to intelligent information filtering: Model, system, and evaluation. *ACM Transactions on Information Systems, Vol. 15, No. 4.* (October 1997), 368–399.
- [PV99] PRAKKEN, H., AND VREESWIJK, G. Logics for Defeasible Argumentation. In *Handbook of Philosophical Logic*, D. Gabbay, Ed. Kluwer Academic Publisher, 1999.
- [Ras92] RASMUSSEN, E. Clustering algorithms. In *Information Retrieval*, W. Frakes and R. Baeza-Yates, Eds. Prentice Hall, 1992, pp. 419–442.
- [RR95] RAO, V., AND RAO, H. *C++ Neural Networks and Fuzzy Logic, Second Edition*. MIS Press, 1995.
- [Ska96] SKAPURA, D. *Building Neural Networks*. ACM Press, Addison-Wesley, 1996.
- [SL92] SIMARI, G. R., AND LOUI, R. P. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence 53* (1992), 125–157.
- [Was89] WASSERMAN, P. D. *Neural Computing. Theory and Practice*. Van Nostrand Reinhold, 1989.
- [ZSS97] ZELLER, M., SHARMA, R., AND SCHULTEN, K. Motion planning of a pneumatic robot using a neural network. *IEEE Control Systems* (June 1997).