An Argumentation Framework for Concept Learning

Leila Amgoud and **Mathieu Serrurier**
IRIT - CNRS
118, route de Narbonne, 31062 Toulouse, France
{amgoud, serrurier}@irit.fr

Abstract

Concept learning is an important problem in AI that consists of, given a set of training examples and counter-examples on a particular concept, identifying a model that is coherent with the training examples, i.e that classifies them correctly. The obtained model is intended to be reused for the purpose of classifying new examples. Version space is one of the *formal* frameworks developed for that purpose. It takes as input a consistent set of training examples on the concept to learn, a set of possible models, called *hypothesis* ordered by generality, and returns the hypothesis that are coherent with the training examples. The returned set of hypothesis is called *version space* and is described by its lower and upper bounds.

This paper provides an argumentation-based framework that captures the results of the version space approach. The basic idea is to construct arguments in favor of/against each hypothesis and training example, to evaluate those arguments and to determine among the conflicting arguments the acceptable ones. We will show that the acceptable arguments characterize the version space as well as its lower and upper bounds. Moreover, we will show that an argumentation-based approach for learning offers an additional advantage by allowing the handling of common problems in classical concept learning. Indeed, it is possible to reason directly with sets of hypothesis rather than one, and to deal with inconsistent sets of training examples. Lastly, the framework translates the problem of classifying examples into a decision one.

Introduction

Machine learning aims at building *models* that describe a *concept* from a set of *training examples*. The models are intended to be sufficiently general in order to be reused on new examples. When teexhe concept to learn is binary, i.e. examples of that concept can be either true or false, the problem is called *concept learning*. In (Mitchell 1982), Mitchel has proposed the famous general and abstract framework, called *version space learning*, for concept learning. That framework takes as input a consistent set of *training examples* on the concept to learn, a set of possible models, called *hypothesis* ordered by *generality*, and returns the hypothesis that are coherent with the training examples. The returned set of hypothesis is called *version space* and is described by its lower and upper bounds.

Besides, argumentation has become an Artificial Intelligence keyword for the last fifteen years, especially in sub-fields such as non monotonic reasoning, inconsistency-tolerant reasoning, multiple-source information systems (Amgoud & Cayrol 2002; Prakken & Sartor 1997; Simari & Loui 1992; Gómez & Chesñevar 2003). Argumentartion follows three steps: i) to construct arguments and counter-arguments for a statement, ii) to select the "acceptable" ones and, finally, iii) to determine whether the statement can be accepted or not.

This paper claims that argumentation can also be used as an alternative approach for concept learning, which not only retrieves in an elegant way the results of the version space learning framework, but also offers several other advantages. In this argumentation-based approach, the concept learning problem is reformulated as follows: given a set of examples (the training ones, and/or additional examples) and a set of hypothesis, what should be the class of a given example? To answer this question, arguments are constructed in favor of all the possible classifications of the examples. A classification can come either from an hypothesis or from a training example. The obtained arguments may be conflicting since it may be the case that the same example is affected to different classes. We will show that the acceptability semantics defined in (Dung 1995) allow us to identify and characterize the version space as well as its lower and upper bounds.

The framework presents also the following features that make it original and flexible:

1. it handles i) the case of a consistent set of training examples; ii) the case of an inconsistent set of training examples; and iii) the case of an empty set of training examples;
2. it allows one to reason directly on the set of hypothesis;
3. examples are classified on the basis of the whole set of hypothesis rather than only one hypothesis as it is the case in standard concept learning. Indeed, in the standard approach, a unique hypothesis is chosen, and all the examples are classified on the basis of that hypothesis.
4. it presents different original and intuitive decision criteria for choosing the class of an example.

The paper is organized as follows. We first present the version space learning framework, then we introduce the basic argumentation framework of Dung (Dung 1995). The third section introduces our argumentation-based framework for learning.

Vesion Space Learning

The aim of this section is to introduce the version space framework developed by Mitchel in (Mitchell 1982). Let $\mathcal{X}$ denote a *feature space* used for describing examples. Elements of $\mathcal{X}$ may then be pairs (attribute, value), first order facts, ... This set $\mathcal{X}$ is supposed to be equipped with an equivalence relation $\equiv$. Let $\mathcal{U}$ = $\{0,1\}$ be a *concept space*, where 1 means that the example of the concept is positive, and 0 means that the example is negative.
The version space framework takes as input a *hypothesis space* $\mathcal{H}$, and a set $\mathcal{S}$ of m *training examples*.

$$\mathcal{S} = \{(x_i, u_i)_{i=1,\ldots,m} \text{ s.t } x_i \in \mathcal{X} \text{ and } u_i \in \mathcal{U}\}$$

Note that the set $\mathcal{S}$ contains both positive examples (i.e. the value of $x - i$ is equal to 1), and negative ones (i.e. the value of $x - i$ is equal to 0). Otherwise, the learning problem becomes trivial and not genuine. An important notion in concept learning is that of consistency. In fact, a set of examples is said to be consistent if it does not contain two logically equivalent examples with two different values. Formally:

Definition 1 (Consistency) *The set $\mathcal{S}$ of training examples is* consistent *iff* $\nexists$ *(x_1, u_1), $(x_2, u_2) \in \mathcal{S}$ such that $x_1 \equiv x_2$ and $u_1 \neq u_2$. Otherwise, $\mathcal{S}$ is said* inconsistent*.*

Regarding the *hypothesis space* $\mathcal{H}$, it may be, for instance, decision trees, propositional sets of rules, neural nets ... An *hypothesis* h is a mapping from $\mathcal{X}$ to $\mathcal{U}$ (i.e. h: $\mathcal{X} \mapsto \mathcal{U}$). The set $\mathcal{H}$ is supposed to be equipped with a *partial preorder* $\succeq$ that reflects the idea that some hypothesis are more general than others in the sense that they classify positively more examples. This preorder defines a lattice on the hypothesis space. Formally:

Definition 2 (Generality order on hypothesis) *Let h_1, h_2 $\in \mathcal{H}$. h_1 is* more general *than h_2, denoted by $h_1 \succeq h_2$, iff $\{x \in \mathcal{X} | h_1(x) = 1\} \supseteq \{x \in \mathcal{X} | h_2(x) = 1\}$.*

Before defining the output of the framework, let us first introduce a key notion, that of *soundness*.

Definition 3 (Soundness) *Let $h \in \mathcal{H}$. An hypothesis h is* sound *with respect to a training example $(x, u) \in \mathcal{S}$ iff $h(x) = u$. If $\forall (x_i, u_i) \in \mathcal{S}$, h is sound w.r.t (x_i, u_i), then h is said to be* sound *with $\mathcal{S}$.*

The framework identifies the *version space*, which is the set $\mathcal{V}$ of all the hypothesis of $\mathcal{H}$ that are sound with $\mathcal{S}$. The idea is that a "good" hypothesis should at least classify the training examples correctly.

Definition 4 (Version space)

$$\mathcal{V} = \{h \in \mathcal{H} | \text{ h is sound with } \mathcal{S}\}$$

Version space learning aims at identifying the *upper* and the *lower* bounds of this version space $\mathcal{V}$. The upper bound will contain the most general hypothesis, i.e the ones that classify more examples, whereas the lower bound will contain the most specific ones, i.e the hypothesis that classify less examples.

Definition 5 (General hypothesis) *The set of* general hypothesis *is $\mathcal{V}_G$ = $\{h \in \mathcal{H} \mid h$ is sound with $\mathcal{S}$ and $\nexists\, h' \in \mathcal{H}$ with h' sound with $\mathcal{S}$, and $h' \succeq h\}$.*

Definition 6 (Specific hypothesis) *The set of* specific hypothesis *is $\mathcal{V}_S$ = $\{h \in \mathcal{H} \mid h$ is sound with $\mathcal{S}$ and $\nexists\, h' \in \mathcal{H}$ with h' sound with $\mathcal{S}$, and $h \succeq h'\}$.*

From the above definition, we have the following simple property characterizing the elements of $\mathcal{V}$.

Property 1 *(Mitchell 1982)*

$$\mathcal{V} = \{h \in \mathcal{H} | \exists h_1 \in \mathcal{V}_S, \exists h_2 \in \mathcal{V}_G, h_2 \succeq h \succeq h_1\}$$

In (Mitchell 1982), an algorithm that computes the version space $\mathcal{V}$ by identifying its upper and lower bounds $\mathcal{V}_S$ and $\mathcal{V}_G$ has been proposed.

The above framework has some limits. First, finding the version space is not sufficient for classifying examples out of the training set. This is due to possible conflicts between hypothesis. Second, it has been shown that the complexity of the algorithm that identifies $\mathcal{V}_S$ and $\mathcal{V}_G$ is very high. In order to palliate that limit, learning algorithms try in general to reach only one hypothesis in the version space by using heuristical exploration of $\mathcal{H}$ (from general to specific exploration, for instance FOIL (Quinlan 1990), or from specific to general exploration, for instance PROGOL (Muggleton 1995)). That hypothesis is then used for classifying new objects. Moreover, it is obvious that this framework does not support inconsistent set of examples:

Property 2 *(Mitchell 1982) If the set $\mathcal{S}$ is inconsistent, then the version space $\mathcal{V}$ = $\emptyset$.*

A consequence of the above result is that no concept can be learned. This problem may appear in the case of noisy training data set.

Let us illustrate the above definitions through the following example in which we try to learn the concept "a sunny day".

Example 1 (Learning the concept sunny day) *In this example, the feature space is a pair (attribute, value). Three attributes are considered: pressure, temperature, and humidity. Four training examples are given, and are summarized in Table below. For instance (pressure, low), (temperature, medium), and (humidity, high) is a negative example for the concept a sunny day, whereas the (pressure, medium), (temperature, medium), and (humidity, low) is a positive one.*

pressure	*temperature*	*humidity*	*sunny*
low	*medium*	*high*	*0*
medium	*medium*	*low*	*1*
low	*medium*	*medium*	*0*
medium	*high*	*medium*	*1*

Let us suppose that the hypothesis space $\mathcal{H}$ is the space of constrains *on the values of each attribute. Indeed, the constraints are conjunctions of accepted values of attributes.*

The special constraint $\emptyset$ (resp. $?$) means that no (resp. all) values of attributes are accepted. If a vector of values of attributes match all the constraints, then it is considered as a positive *example, otherwise it is a* negative *one. The hypothesis $\langle \emptyset, \emptyset, \emptyset \rangle$ and $\langle ?, ?, ? \rangle$ are respectively the lower and the upper bound of the hypothesis space $\mathcal{H}$. Using the version space learning algorithm, we get: $\mathcal{V}_G$ = {⟨ medium ∨ high, ?, ?⟩} and $\mathcal{V}_S$ = {⟨ medium, medium ∨ high, low ∨ medium⟩}. Here $\mathcal{V}_S$ and $\mathcal{V}_G$ contain both only one hypothesis. The hypothesis in V_G, for instance, considers as positive examples of the sunny day concept all features that have $medium$ or $high$ values for the pressure attribute.*

Abstract Argumentation Framework

Argumentation is a reasoning model that follows the following steps:

1. Constructing *arguments* and counter-arguments.
2. Defining the *strengths* of those arguments.
3. Evaluating the *acceptability* of the different arguments.
4. Concluding or defining the *justified conclusions*.

In (Dung 1995), an argumentation system is defined as follows:

Definition 7 (Argumentation system) *An* argumentation system *(AS) is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$. $\mathcal{A}$ is a set arguments and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is a defeasibility relation. We say that an argument A defeats an argument B iff $(A, B) \in \mathcal{R}$ (or $A \; \mathcal{R} \; B$).*

Note that to each argumentation system is associated an oriented graph whose nodes are the different arguments, and the edges represent the defeasibility relationship between them. Among all the conflicting arguments, it is important to know which arguments to keep for inferring conclusions or for making decisions. In (Dung 1995), different semantics for the notion of acceptability have been proposed. Let's recall them here.

Definition 8 (Conflict-free, Defence) *Let $\mathcal{B} \subseteq \mathcal{A}$.*

- *$\mathcal{B}$ is* conflict-free *iff there exist no A_i, $A_j \in \mathcal{B}$ such that $A_i \; \mathcal{R} \; A_j$.*
- *$\mathcal{B}$* defends *an argument A_i iff for each argument $A_j \in \mathcal{A}$, if $A_j \; \mathcal{R} \; A_i$, then there exists $A_k \in \mathcal{B}$ such that $A_k \; \mathcal{R} \; A_j$.*

Definition 9 (Acceptability semantics) *Let $\mathcal{B}$ be a conflict-free set of arguments, and let $\mathcal{F}$: $2^{\mathcal{A}} \mapsto 2^{\mathcal{A}}$ be a function such that $\mathcal{F}(\mathcal{B})$ = $\{A \mid \mathcal{B}$ defends $A\}$.*

- *$\mathcal{B}$ is a* complete extension *iff $\mathcal{B}$ = $\mathcal{F}(\mathcal{B})$.*
- *$\mathcal{B}$ is a* grounded extension *iff it is the minimal (w.r.t. set-inclusion) complete extension.*
- *$\mathcal{B}$ is a* preferred extension *iff it is a maximal (w.r.t. set-inclusion) complete extension.*
- *$\mathcal{B}$ is a* stable extension *iff it is a preferred extension that defeats all arguments in $\mathcal{A} \backslash \mathcal{B}$.*

Let $\{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$ be the set of all possible extensions under a given semantics.

Note that there is only one grounded extension which may be empty. It contains all the arguments which are not defeated, and also the arguments which are defended directly or indirectly by non-defeated arguments.
The last step of an argumentation process consists of determining, among all the conclusions of the different arguments, the "good" ones, called *justified conclusions*.

An Argumentation Framework for Concept Learning

The aim of this section is to propose an instantiation of the general and abstract framework of Dung that allows learning concepts from sets of trainng examples. We will show that this argumentation-based model captures the results of the version space learning presented in a previous section. The sets of *version space*, *specific* and *general* hypothesis are characterized in our model. Since the classical approach of version space learning considers only the case where the set of training examples is consistent, we will present two versions of our model. In the first one, the set $\mathcal{S}$ is supposed to be consistent. This model is then generalized to the case where $\mathcal{S}$ can be inconsistent. We will show that even in this latter case, the version space $\mathcal{V}$ is not always empty, thus it is still possible to learn concepts.

Throughout this section, we will consider a features space $\mathcal{X}$, a concept space $\mathcal{U} = \{0, 1\}$, a hypothesis space $\mathcal{H}$, which is equipped with a partial preordering $\succeq$ (see Definition 2), and a set $\mathcal{S}$ of $m > 0$ training examples.

Consistent Case

In order to instantiate the abstract framework of Dung, one needs to define the set $\mathcal{A}$ or arguments as well as the defeasibility relationship between those arguments. In our particular application, one needs to argue about particular classifications, thus arguments are constructed in favor of assigning particular values from $\mathcal{U}$ to an example in $\mathcal{X}$. Indeed, an argument in favor of the pair (x, u) represents the reason of assigning the value u to the example x. Two reasons can be distinguished:

1. (x, u) is a training example in $\mathcal{S}$,
2. there exists a hypothesis $h \in \mathcal{H}$ that classifies x in u.

Definition 10 (Argument) *An* argument *is a triplet A = $\langle h, x, u \rangle$ such that:*

1. *$h \in \mathcal{H}$, $x \in \mathcal{X}$, $u \in \mathcal{U}$*
2. *If $h \neq \emptyset$, then u = $h(x)$*
3. *If h = $\emptyset$, then $(x, u) \in \mathcal{S}$*

h is called the support *of the argument, and (x, u) its* conclusion. *Let* `Example`(A) = x, *and* `Value`(A) = u.
Let $\mathcal{A}$ be the set of arguments built from $(\mathcal{H}, \mathcal{X}, \mathcal{U})$.

Note that from the above definition for any trainng example $(x_i, u_i) \in \mathcal{S}$, $\exists \langle \emptyset, x_i, u_i \rangle \in \mathcal{A}$. Let $\mathcal{A}_{\mathcal{S}}$ = $\{\langle \emptyset, x, u \rangle \in \mathcal{A}\}$ (i.e. the set of arguments coming from the training examples).

Property 3 *Let $\mathcal{S}$ be a set of training examples. $|\mathcal{S}| = |\mathcal{A}_\mathcal{S}|$*[1].

Proof *This follows from the above definition, and from the fact that a hypothesis h cannot be empty.* ■

Since the set $\mathcal{S}$ of training examples is not empty, then $\mathcal{A}_\mathcal{S}$ is not empty as well.

Property 4 $\mathcal{A}_\mathcal{S} \neq \emptyset$.

Proof *This follows directly from the above property, i.e. $|\mathcal{S}| = |\mathcal{A}_\mathcal{S}|$, and the assumption that $\mathcal{S} \neq \emptyset$.* ■

Let us illustrate the notion of argument through example 1.

Example 2 *In example 1, there are exactly four arguments with an empty support, and they correspond to the training examples: $\mathcal{A}_\emptyset = \{a_1 = \langle\emptyset$, (pressure, low) $\wedge$ (temperature, medium) $\wedge$ (humidity,high), 0$\rangle$,*
$a_2 = \langle\emptyset$, (pressure, medium) $\wedge$ (temperature, medium) $\wedge$ (humidity, low), 1$\rangle$,
$a_3 = \langle\emptyset$, (pressure, low) $\wedge$ (temperature, medium) $\wedge$ (humidity, medium), 0$\rangle$,
$a_4 = \langle\emptyset$, (pressure, medium) $\wedge$ (temperature, high) $\wedge$ (humidity, medium), 1$\rangle\}$. There are also arguments with a non-empty support such as:
$\langle a_5 = \langle$? , medium $\vee$ high, ?$\rangle$, (pressure, low) $\wedge$ (temperature, high) $\wedge$ (humidity, high), 1$\rangle\}$,
$a_6 = \langle\langle$ medium$\vee$ high, ?, ?$\rangle$, (pressure, low) $\wedge$ (temperature, high) $\wedge$ (humidity, high), 0$\rangle$,
$a_7 = \langle\langle$ medium, medium$\vee$ high, ?$\rangle$, (pressure, low) $\wedge$ (temperature, high) $\wedge$ (humidity, high), 0$\rangle$.

In (Amgoud & Cayrol 2002; Prakken & Sartor 1997; Simari & Loui 1992), it has been argued that arguments may have different strengths depending on the quality of information used to construct them. In (Simari & Loui 1992), for instance, arguments built from specific information are stronger than arguments built from more general ones. In our particular application, it is clear that arguments with an empty support are stronger than arguments with a non-empty one. This reflects the fact that classifications given by training examples take precedence over ones given by hypothesis in $\mathcal{H}$. It is also natural to consider that arguments using more general hypothesis are stronger than arguments with less general hypothesis.

Definition 11 (Comparing arguments) *Let $\langle h, x, u\rangle$, $\langle h', x', u'\rangle$ be two arguments of $\mathcal{A}$. $\langle h, x, u\rangle$ is* preferred *to $\langle h', x', u'\rangle$, denoted by $\langle h, x, u\rangle$* Pref *$\langle h', x', u'\rangle$, iff:*

- *$h = \emptyset$ and $h' \neq \emptyset$, or*
- *$h \succeq h'$.*

Property 5 *The relation Pref is a partial preorder.*

Proof *This is due to the fact that the relation $\succeq$ is a partial preorder.* ■

Now that the set of arguments is built, it is possible to define the defeasibility relation $\mathcal{R}$ between arguments in $\mathcal{A}$. Here again, there are two ways in which an argument A can attack another argument B:

1. by *rebutting* its *conclusion*. This situation occurs when the two arguments have contradictory conclusions, i.e. the same example is classified in different ways.
2. by *undercutting* its *support*. This occurs when the support of B classifies in a different way the example of the conclusion of A. However, this relation is only restricted to training examples. Indeed, only arguments built from training examples are allowed to undercut other arguments. The idea behind this is that training examples are the only, in some sense, certain information one has, and thus cannot be defeated by hypothesis. However, hypothesis have controversial status.

Definition 12 (Rebutting) *Let $\langle h, x, u\rangle$, $\langle h', x', u'\rangle$ be two arguments of $\mathcal{A}$. $\langle h, x, u\rangle$* rebuts *$\langle h', x', u'\rangle$ iff $x \equiv x'$, $u \neq u'$.*

Example 3 *In example 2, we have for instance :*
a_5 rebuts a_6*, a_5* rebuts a_7*, a_6* rebuts a_5*, and a_7* rebuts a_5.

Definition 13 (Undercutting) *Let $\langle h, x, u\rangle$, $\langle h', x', u'\rangle$ be two arguments of $\mathcal{A}$. $\langle h, x, u\rangle$* undercuts *$\langle h', x', u'\rangle$ iff $h = \emptyset$ and $h'(x) \neq u$.*

Example 4 *In example 2, we have for instance :*
a_1 undercuts a_5*, a_2* undercuts a_5*, a_3* undercuts a_5*, and a_4* undercuts a_5.

The rebutting and undercutting relations are used in most argumentation systems that handle inconsistency in knowledge bases.

Property 6 *If $\mathcal{S}$ is consistent, then $\nexists$ A, $B \in \mathcal{A}_\mathcal{S}$ such that A rebuts B, or A undercuts B.*

Proof *Let $A = \langle\emptyset, x, u\rangle$, $B = \langle\emptyset, x', u'\rangle \in \mathcal{S}$ such that A rebuts B. According to Definition 12, $x \equiv x'$ and $u \neq u'$. This contradicts the fact that $\mathcal{S}$ is consistent.* ■

The two above conflict relations are brought together in a unique relation, called "Defeat".

Definition 14 (Defeat) *Let $A = \langle h, x, u\rangle$, $B = \langle h', x', u'\rangle$ be two arguments of $\mathcal{A}$. A* defeats *B iff:*

1. *A rebuts (resp. undercuts) B, and*
2. *(A Pref B), or (not(A Pref B) and not(B Pref A))*

Example 5 *With the argument defined in ex. 2 we have for instance :*
a_1 defeats a_5, a_2 defeats a_5, a_3 defeats a_5, a_4 defeats a_5, a_5 defeats a_6, a_5 defeats a_7 *and* a_6 defeats a_5.

From the above definition, it is easy to check that an argument with a empty support cannot be defeated by an argument with a non-empty support.

Property 7 *$\forall$ $A \in \mathcal{A}_\mathcal{S}$, $\nexists$ $B \in \mathcal{A} \backslash \mathcal{A}_\mathcal{S}$ s.t B defeats A.*

Proof *Let $A \in \mathcal{A}_\mathcal{S}$ and $B \in \mathcal{A} \backslash \mathcal{A}_\mathcal{S}$ such that B defeats A. This means that B rebuts A (because according to Definition 13, an argument with a non-empty support cannot undercut an argument with an empty one. Moreover, according to Definition 14, we have either B Pref A, or (not(B Pref A) and not(A Pref B)). This is impossible because according to Definition 11, arguments in $\mathcal{A}_\mathcal{S}$ are always preferred to arguments with a non-empty support.* ■

[1]|| denotes the cardinal of a given set

The argumentation system for concept learning is then the following:

Definition 15 (Argumentation system) *An* argumentation system *for concept learning (ASCL) is a pair $\langle \mathcal{A}, defeat \rangle$, where $\mathcal{A}$ is the set of arguments (see Definition 10) and defeat is the relation defined in Definition 14.*

Let us now identify the acceptable arguments of the above ASCL. It is clear that the arguments that are not defeated at all will be acceptable. Let $\mathcal{C}$ denote that set of non-defeated arguments.

Proposition 1 *If $\mathcal{S}$ is consistent, then $\mathcal{A}_\mathcal{S} \subseteq \mathcal{C}$.*

Proof *Let $A \in \mathcal{A}_\mathcal{S}$. Let us assume that $\exists B \in \mathcal{A}$ such that B defeats A. According to Property 7, $B \notin \mathcal{A} \backslash \mathcal{A}_\mathcal{S}$. Thus, $B \in \mathcal{A}_\mathcal{S}$. Moreover, B defeats A means that B rebuts A. This means then that A classifies a training example in u, and B classifies an equivalent example in $u' \neq u$. This contradicts the fact that the set $\mathcal{S}$ is consistent.* ■

From the above proposition and Property 4, it is clear that the ASCL has a non-empty grounded extension.

Proposition 2 (Grounded extension) *If $\mathcal{S}$ is consistent, then the argumentation system $\langle \mathcal{A}, defeat \rangle$ has a non empty grounded extension $\mathcal{E}$.*

Proof *This is due to the fact that $\mathcal{A}_\mathcal{S} \neq \emptyset$ and $\mathcal{A}_\mathcal{S} \subseteq \mathcal{C}$.* ■

Note that the system $\langle \mathcal{A}, defeat \rangle$ is not always finite. By finite we mean that each argument is defeated by a finite number of arguments. This is due to the fact that $\mathcal{H}$ and $\mathcal{X}$ are not always finite.

Proposition 3 *If $\mathcal{H}$ and $\mathcal{X}$ are finite, then the system $\langle \mathcal{A}, defeat \rangle$ is finite.*

When an argumentation system is finite, its characteristic function $\mathcal{F}$ is continuous. Consequently, the least fixed point of this function can be defined by an iterative application of $\mathcal{F}$ to the empty set.

Proposition 4 *If the argumentation system $\langle \mathcal{A}, defeat \rangle$ is finite, then the grounded extension $\mathcal{E}$ is:*

$$\mathcal{E} = \bigcup \mathcal{F}^{i \geq 0}(\emptyset) = \mathcal{C} \cup [\bigcup_{i \geq 1} \mathcal{F}^i(\mathcal{C})].$$

Let us now analyze the other acceptability semantics, namely preferred and stable ones. From Proposition 2, one concludes that the ASCL $\langle \mathcal{A}, defeat \rangle$ has at least one non-empty preferred extensions.

Proposition 5 *If $\mathcal{S}$ is consistent, then the ASCL $\langle \mathcal{A}, defeat \rangle$ has $n \geq 1$ non-empty preferred extensions.*

Proof *In (Dung 1995), it has been shown that the grounded extension is included in very preferred extension. Since the grounded extension is not empty (according to Proposition 2, then there exists at least one non-empty preferred extension).* ■

In general, the preferred extensions of an argumentation system are not stable. However, in our ASCL these extensions coincide. This result is due to the fact that the oriented graph associated to the above ASCL has no odd length circuits. However, it may contain circuits of even length.

Proposition 6

- *The graph associated with the system $\langle \mathcal{A}, defeat \rangle$ has no odd length circuits.*
- *The preferred extensions and stable extensions of the system $\langle \mathcal{A}, defeat \rangle$ coincide.*

Proof (Sketch of the proof) ***Part 1:*** *Let A, B, C be three arguments such that A defeats B, B defeats C, and C defeats A.*

Case 1: *Let us suppose that $A \in \mathcal{A}_\mathcal{S}$.*
According to Property 6, $B \in \mathcal{A} \backslash \mathcal{A}_\mathcal{S}$. According to Property 7, C should be in $\mathcal{A} \backslash \mathcal{A}_\mathcal{S}$. Contradiction because according to Property 7, C cannot defeat A, which is in $\mathcal{A}_\mathcal{S}$.

Case 2: *Let us suppose that $A, B, C \in \mathcal{A} \backslash \mathcal{A}_\mathcal{S}$. This means that A rebuts B, B rebuts C, and C rebuts A (according to Definition 13). Consequently,* $\texttt{Example}(A) \equiv \texttt{Example}(B) \equiv \texttt{Example}(C)$, *and* $\texttt{Value}(A) \neq \texttt{Value}(B)$, $\texttt{Value}(B) \neq \texttt{Value}(C)$. *Due to the fact that $\mathcal{U} = \{0, 1\}$, we have* $\texttt{Value}(A) = \texttt{Value}(C)$. *This contradicts the assumption that C rebuts A.*

Part 2: *This is a consequence of the fact that there is no odd circuits in the system.* ■

Note, however, that the intersection of all the preferred (stable) extensions coincides with the grounded extension.

Proposition 7 *Let $\langle \mathcal{A}, defeat \rangle$ be a ASCL. Let $\mathcal{E}$ be its grounded extension, and $\mathcal{E}_1, \ldots, \mathcal{E}_n$ its preferred (stable) extensions. $\mathcal{E} = \bigcap_{i=1,\ldots,n} \mathcal{E}_i$.*

Let us now show how the above ASCL can retrieve the results of the version space learning, namely the version space and its lower and upper bounds. Before doing that, we start first by introducing some useful notations.

Let `Hyp` be a function that returns for a given set of arguments, their non empty supports. In other words, this function returns all the hypothesis used to build arguments:

Definition 16 *Let $T \subseteq \mathcal{A}$.*

$$\texttt{Hyp}(T) = \{h \mid \exists \langle h, x, u \rangle \in T \text{ and } h \neq \emptyset\}$$

Now we will show that the argumentation-based model for concept learning computes in an elegant way the version space $\mathcal{V}$ (see Definition 4).

Proposition 8 *Let $\langle \mathcal{A}, defeat \rangle$ be a ASCL. Let $\mathcal{E}$ be its grounded extension, and $\mathcal{E}_1, \ldots, \mathcal{E}_n$ its preferred (stable) extensions. If the set $\mathcal{S}$ is consistent then:*

$$\texttt{Hyp}(\mathcal{E}) = \texttt{Hyp}(\mathcal{E}_1) = \ldots = \texttt{Hyp}(\mathcal{E}_n) = \mathcal{V}$$

where $\mathcal{V}$ is the version space.

Proof *Let $\mathcal{E}_i$ be an extension under a given semantics.*

$\texttt{Hyp}(\mathcal{E}_i) \subseteq \mathcal{V}$**:** *Let $h \in \texttt{Hyp}(\mathcal{E}_i)$, then $\exists \langle h, x, u \rangle \in \mathcal{E}_i$.*
Let us assume that $\exists (x_i, u_i) \in \mathcal{S}$ such that $h(x_i) \neq u_i$. This means $\langle \emptyset, x_i, u_i \rangle$ undercuts $\langle h, x, u \rangle$ (according to Definition 13). Consequently, $\langle \emptyset, x_i, u_i \rangle$ defeats $\langle h, x, u \rangle$. However, according to Property 3, $\langle \emptyset, x_i, u_i \rangle \in \mathcal{A}_\mathcal{S}$, thus $\langle \emptyset, x_i, u_i \rangle \in \mathcal{E}_i$. Contradiction because $\mathcal{E}_i$ is an extension, thus by definition it is conflict-free.

$\mathcal{V} \subseteq \mathtt{Hyp}(\mathcal{E}_i)$: *Let $h \in \mathcal{V}$, and let us assume that $h \notin \mathtt{Hyp}(\mathcal{E}_i)$. Since $h \in \mathcal{V}$, then $\forall (x_i, u_i) \in \mathcal{S}$, $h(x_i) = u_i$ (1) Let $(x, u) \in \mathcal{S}$, thus $h(x) = u$ and consequently $\langle h, x, u\rangle \in \mathcal{A}$. Moreover, since $h \notin \mathtt{Hyp}(\mathcal{E})$, then $\langle h, x, u\rangle \notin E$. Thus, $\exists$ $\langle h', x', u'\rangle$ that defeats $\langle h, x, u\rangle$.*

- *Case 1: $h' = \emptyset$. This means that $\langle \emptyset, x', u'\rangle$ undercuts $\langle h, x, u\rangle$ and $h(x') \neq u'$ Contradiction with (1).*
- *Case 2: $h' \neq \emptyset$. This means that $\langle h', x', u'\rangle$ rebuts $\langle h, x, u\rangle$. Consequently, $x \equiv x'$ and $u \neq u'$. However, since $h \in \mathcal{V}$, then h is sound with $\mathcal{S}$. Thus, $\langle \emptyset, x, u\rangle$ defeats $\langle h', x', u'\rangle$, then $\langle \emptyset, x, u\rangle$ defeats $\langle h, x, u\rangle$. Since $\langle \emptyset, x, u\rangle \in \mathcal{S}$, then $\langle h, x, u\rangle \in \mathcal{F}(\mathcal{C})$ and consequently, $\langle h, x, u\rangle \in \mathcal{E}_i$. Contradiction.*

■

The above result is of great importance. It shows that to get the version space, one only needs to compute the grounded extension.
We can also show that if a given argument is is an extension $\mathcal{E}_i$, then any argument based on an hypothesis from the version space that supports the same conclusion is in that extension. Formally:

Proposition 9 *Let $\langle \mathcal{A}, defeat\rangle$ be a ASCL, and $\mathcal{E}_1, \ldots, \mathcal{E}_n$ its extensions under a given semantics. If $< h, x, u > \in \mathcal{E}_i$, then $\forall h' \in \mathcal{V}$ s.t. $h' \neq h$ if $h'(x) = u$ then $< h', x, u > \in \mathcal{E}_i$.*

Proof *Let $\mathcal{E}_i$ be a given extension, and let $\langle h, x, u\rangle \in \mathcal{E}_i$. Let $h' \in \mathcal{V}$ such that $h'(x) = u$. Let us assume that $\langle h', x, u\rangle \notin \mathcal{E}_i$.*

Case 1: *$\mathcal{E}_i \cup \{\langle h', x, u\rangle\}$ is not conflict-free. This means that $\exists$ $\langle h", x", u"\rangle \in \mathcal{E}_i$ such that $\langle h", x", u"\rangle$ defeats $\langle h', x, u\rangle$. Consequently, $\langle h", x", u"\rangle$ undercuts $\langle h', x, u\rangle$ if $h" = \emptyset$, or $\langle h", x", u"\rangle$ rebuts $\langle h', x, u\rangle$ if $h" \neq \emptyset$.*
If $h" = \emptyset$, then $h'(x") \neq u"$, this contradicts the fact that $h' \in \mathcal{V}$.
If $h" \neq \emptyset$, then $x" \equiv x$ and $u" \neq u$ and either $h" \succeq h'$, or $h', h"$ are not comparable. Thus, $\langle h", x", u"\rangle$ rebuts $\langle h, x, u\rangle$. Since $\langle h, x, u\rangle, \langle h", x", u"\rangle \in \mathcal{E}_i$, then h and $h"$ are not comparable. But, this means that $\langle h, x, u\rangle$ defeats $\langle h", x", u"\rangle$, and $\langle h", x", u"\rangle$ defeats $\langle h, x, u\rangle$. Consequently, $\mathcal{E}_i$ is not conflict-free. Contradiction because $\mathcal{E}_i$ is an extension.

Case 2: *$\mathcal{E}_i$ does not defend $\langle h', x, u\rangle$. This means that $\exists$ $\langle h", x", u"\rangle$ defeats $\langle h', x, u\rangle$.*

- *Case 1: $h" = \emptyset$. This means that $h'(x") \neq u"$. Contradiction because $h' \in \mathcal{V}$.*
- *Case 2: $h" \neq \emptyset$. This means that $x \equiv x"$, $u \neq u"$, and $h" \succeq h'$. Thus, $\langle h", x", u"\rangle$ rebuts $\langle h, x, u\rangle$.*
  If $h \succeq h"$, then $\langle h, x, u\rangle$ defeats $\langle h", x", u"\rangle$, thus $\langle h, x, u\rangle$ defends $\langle h', x, u\rangle$.
  If $h" \succeq h$, then $\langle h", x", u"\rangle$ defeats $\langle h, x, u\rangle$. However, since $\langle h, x, u\rangle \in \mathcal{E}$, then $\mathcal{E}$ defends $\langle h, x, u\rangle$ against $\langle h", x", u"\rangle$. Thus, $\mathcal{E}$ defends $\langle h', x, u\rangle$. Contradiction

■

Using the grounded extension, one can characterize the upper and the lower bounds of the version space. The upper bound corresponds to the most preferred w.r.t Pref arguments of the grounded extension, whereas the lower bound corresponds to the less preferred ones.

Proposition 10 *Let $\langle \mathcal{A}, defeat\rangle$ be a ASCL, and $\mathcal{E}$ its grounded extension.*

- *$\mathcal{V}_G = \{h \mid \exists <h, x, u> \in \mathcal{E}$ s.t $\forall <h', x', u'> \in \mathcal{E}$, not $(<h', x', u'>$ Pref $<h, x, u>)\}$.*
- *$\mathcal{V}_S = \{h \mid \exists <h, x, u> \in \mathcal{E}$ s.t $\forall <h', x', u'> \in \mathcal{E}$, not $(<h, x, u>$ Pref $<h', x', u'>)\}$.*

Proof
$\mathcal{V}_G = \{h \mid \exists <h, x, u> \in \mathcal{E}$ s.t $\forall <h', x', u'> \in \mathcal{E}$, not $(<h', x', u'>$ Pref $<h, x, u>)\}$.

- *Let $h \in \mathcal{V}_G$, thus $h \in \mathcal{V}$, and $\forall h' \in \mathcal{V}$, $h \succeq h'$. Since $h \in \mathcal{V}$, thus, $h \in \mathtt{Hyp}(\mathcal{E})$, with $\mathcal{E}$ an extension. Then, $\exists \langle h, x, u\rangle \in \mathcal{E}$. Since $h \succeq h'$ for any $h' \in \mathcal{V}$, then $h \succeq h'$ for any $h' \in \mathtt{Hyp}(\mathcal{E})$. Thus, $\langle h, x, u\rangle$ Pref $\langle h', x', u'\rangle$, $\forall$ $\langle h', x', u'\rangle \in \mathcal{E}$.*
- *Let $\langle h, x, u\rangle \in \mathcal{E}$ such that $\forall$ $\langle h', x', u'\rangle \in \mathcal{E}$, and not$(\langle h', x', u'\rangle Pref \langle h, x, u\rangle)$. Thus, $h \in \mathtt{Hyp}(\mathcal{E})$, and $\forall$ $h' \in \mathtt{Hyp}(\mathcal{E})$, not$(h' \succeq h)$, thus $h \in \mathcal{V}_G$.*

$\mathcal{V}_S = \{h \mid \exists <h, x, u> \in \mathcal{E}$ s.t $\forall <h', x', u'> \in \mathcal{E}$, not $(<h, x, u>$ Pref $<h', x', u'>)\}$.

- *Let $h \in \mathcal{V}_S$, thus $\nexists h' \in \mathcal{V}$ such that $h \succeq h'$. Since $h \in \mathcal{V}_S$, then $h \in \mathcal{V}$ and consequently, $h \in \mathtt{Hyp}(\mathcal{E})$. This means that $\exists \langle h, x, u\rangle \in \mathcal{E}$. Let us assume that $\exists \langle h', x', u'\rangle \in \mathcal{E}$ such that $\langle h, x, u\rangle$ $Pref$ $\langle h', x', u'\rangle$, thus $h \succeq h'$. Contradiction with the fact that $h \in \mathcal{V}_S$.*
- *Let $\langle h, x, u\rangle \in \mathcal{E}$ such that $\forall$ $\langle h', x', u'\rangle \in \mathcal{E}$, and not$(\langle h, x, u\rangle Pref \langle h', x', u'\rangle)$, thus not$(h \succeq h')$. Since $h \in \mathcal{V}$, and $\forall h' \in \mathcal{V}$, not$(h \succeq h')$, then $h \in \mathcal{V}_S$.*

■

As said before, the last step of an argumentation process consists of defining the *status* of the conclusions, in our case, the classification of examples. In what follows we present two decision criteria: The first one, called universal vote, consists of accepting those classifications that are in any extension. However, it is clear that this kind of voting may not classify all the examples. Thus, we propose a second criterion, called majority vote, that allows to associate a class to each example. The conclusions here are the ones that are supported by a majority of arguments that appear in the different extensions. Formally:

Definition 17 *Let $\langle \mathcal{A}, defeat\rangle$ be a ASCL, and $\mathcal{E}_1, \ldots, \mathcal{E}_n$ its extensions under a given semantics. Let $x \in \mathcal{X}$ and $u \in \mathcal{U}$.*

Universal vote: *x is* universally classified *in u iff $\forall \mathcal{E}_i$, $\exists$ $<h, x, u> \in \mathcal{E}_i$. UV denotes the set of all universally classified examples.*

Majority vote: *x is* majoritarily classified *in u iff $|\{< h, x, u > | \exists \mathcal{E}_i, < h, x, u > \in \mathcal{E}_i\}| \geq |\{< h, x, u' > | u' \neq u, \exists \mathcal{E}_i, < h, x, u' > \in \mathcal{E}_i\}|$. MV denotes the set of all majoritarily conclusions accepted by majority vote.*

The universally classified examples are those which are supported by arguments in all the extensions. From a learning point of view, these correspond to examples classified by the most general hypothesis in the version space. Thus, according to Proposition 7, we have the following result:

Property 8 *Let $\langle \mathcal{A}, defeat \rangle$ be a ASCL, and $\mathcal{E}$ its grounded extension :*

$$UV = \{(x,u) | \exists < h, x, u > \in \mathcal{E}\}$$

It is easy to check that the set of universally classified examples is included in the set of majoritarily classified ones.

Property 9 *Let $\langle \mathcal{A}, defeat \rangle$ be a ASCL, and $\mathcal{E}_1, \ldots, \mathcal{E}_n$ its extensions under a given semantics :*

$$UV \subseteq MV$$

Proof *This result follows directly from the fact that the extensions are conflict-free.* ■

Inconsistent Case

Let us now consider the case where $\mathcal{S}$ is inconsistent, with $\mathcal{S}$ can be partionned into $\mathcal{S}_1, \ldots, \mathcal{S}_n$, suth that each $\mathcal{S}_i$ is a maximal (for set inclusion) consistent subsets of $\mathcal{S}$. This means that some training examples are classified in different classes. However, all the elements of $\mathcal{S}$ are supposed to be equally preferred. In this case, two arguments supporting such conflicting training examples rebut each other, thus can defeat each other as well.

Let $\mathcal{A}_{\mathcal{S}_1}$, ..., $\mathcal{A}_{\mathcal{S}_n}$ be the sets of arguments with an empty support and whose conclusions are respectively in the subsets of training examples $\mathcal{S}_1, \ldots, \mathcal{S}_n$. It is clear that each set $\mathcal{A}_{\mathcal{S}_i}$ is conflict-free, however, it is defeated by arguments in $\mathcal{A}_{\mathcal{S}_j}$ with $i \neq j$.

As in the consistent case, arguments with an empty support are preferred to arguments built from hypothesis. In what follows, $ASCL_i$ will denote the argumentation system in the inconsistent case. Note that all the above definitions of an argument, of defeat, do not change. It can be checked that the corresponding oriented graph of $ASCL_i$ does not contain odd length circuits.

Proposition 11

- *The graph associated to the system $ASCL_i$ has no odd length circuits.*
- *In the $ASCL_i$ $\langle \mathcal{A}, defeat \rangle$, preferred extensions and stable ones coincide.*

As a consequence of the above result, the system $ASCL_i$ has preferred extensions that are also stable. Moreover, the grounded extension of this system coincides with the intersection of all the preferred extensions. Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be the preferred extensions of that system, and $\mathcal{E}$ its grounded extension.

Proposition 12 *If $\mathcal{S}$ is inconsistent and non empty, then the $ASCL_i$ $\langle \mathcal{A}, defeat \rangle$ has still $n \geq 1$ non-empty preferred extensions. Moreover, $\mathcal{E} = \bigcap_{i=1,\ldots,n} \mathcal{E}_i$.*

However, the grounded extension can be empty in this particular case of inconsistent training examples. The above result is of great importance since it shows that even in this particular case of inconsistent training example, it is still possible to classify examples.

Note that each preferred/stable extension contains one of the sets $\mathcal{A}_{\mathcal{S}_1}$, ..., $\mathcal{A}_{\mathcal{S}_n}$. Moreover, the same set $\mathcal{A}_{\mathcal{S}_i}$ may belong to several extensions at the same time. It can be shown that all the hypothesis that are used to build arguments in a given extension are sound with the subset of training examples of that extension. Indeed, for each consistent subset of $\mathcal{S}$, we get the extensions of the consistent case previously studied.

Case of an Empty Set of Training Examples

Another interesting case is when the set of training examples is empty. In this case, the learning problem consists of classifying examples only on the basis of a set $\mathcal{H}$ of hypothesis. This is, indeed, a particular case of the previous case where $\mathcal{S}$ is inconsistent. The corresponding argumentation system constructs then arguments only on the basis of hypothesis, thus there is no argument with an empty support. This system satisfies exactly the same properties as the $ASCL_i$. For instance, the corresponding graph has no odd length circuits. Moreover, it contains at least one non-empty preferred extension, which is also stable. The grounded extension of this sytstem is the intersection of all the preferred extensions.

Conclusion

This paper has proposed, to the best of our knowledge, the first argumentation-based framework for concept learning. This framework considers the learning problem as a process that follows four main steps: it first constructs arguments in favor of classifications of examples from a set of training examples, and a set of hypothesis. Conflicts between arguments may appear when two arguments classify the same example in different classes. Once the arguments identified, it is possible to compare them on the basis of their strengths. The idea is that arguments coming from the set of training examples are stronger than arguments built from the set of hypothesis. Similarly, arguments based on general hypothesis are stronger than arguments built from more specific hypothesis. Indeed, such preference relation between arguments ensures that during the learning process, only hypothesis that are sound with the training examples are kept, and general hypothesis are privileged to less specific ones. We have shown that acceptability semantics of the ASCL retrieves and even characterizes the version space and its upper and lower bounds. Thus, the argumentation-based approach gives another interpretation of the version space as well as its two bounds in terms of arguments. We have also shown that when the set of training examples is inconsistent, it is still possible to learn concepts, and to classify examples of it. Indeed, in this particular case, the version space is empty as it is the case in the version space learning framework. A last and not least feature of our framework consists of defining the class of each example on the basis of all the hypothesis and not only one, and also to suggest four intuitive decision criteria for that purpose.
A first extension of this framework would be to explore the proof theories in argumentation that test directly whether a given argument is in the grounded extension without com-

puting this last. This means that one may know the class of an example without exploring the whole hypothesis space.

This framework can easily be generalized to the case of classification problems in which an example can be affected to a class among a set of possible classes. It could also be extended in order to handle qualitative uncertainty and preferences on examples. It can also be extended to handle the regression problem in which the concept takes the form of a continuous function.

Acknowledgments

This work was supported by the Commission of the European Communities under contract IST-2004-002307, ASPIC project "Argumentation Service Platform with Integrated Components".

References

Amgoud, L., and Cayrol, C. 2002. Inferring from inconsistency in preference-based argumentation frameworks. *Int. Journal of Automated Reasoning* Volume 29 (2):125–169.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77:321–357.

Gómez, S. A., and Chesñevar, C. I. 2003. Integrating defeasible argumentation with fuzzy art neural networks for pattern classification. In *Proc. ECML'03*.

Mitchell, T. 1982. Generalization as search. *Artificial intelligence* 18:203–226.

Muggleton, S. 1995. Inverse entailment and Progol. *New Generation Computing* 13:245–286.

Prakken, H., and Sartor, G. 1997. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics* 7:25–75.

Quinlan, J. R. 1990. Learning logical definitions from relations. *Machine Learning* 5:239–266.

Simari, G. R., and Loui, R. P. 1992. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence and Law* 53:125–157.