Description Identification and Retraction with Integrated Instance/Concept-Based Boundary Sets

Evgueni N. Smirnov and Peter J. Braspenning

Department of Computer Science, Maastricht University
P.O.BOX 616, 6200 MD Maastricht, The Netherlands
e-mail: {smirnov, braspenning}@cs.unimaas.nl

Abstract

This paper presents incremental version space algorithms for description identification and retracting training data. The correctness of the algorithms is proven for the class of admissible description languages when version spaces to be learned are represented with integrated instance/concept-based boundary sets (Smirnov and Braspenning 1998b). It is shown that the exponential complexity of description identification and retracting data is avoided when generation of version spaces with respect to particular training descriptions is polynomial in the relevant properties of admissible languages.

Introduction

Description identification is a task of acquiring descriptions in description languages L_d that are consistent with training instances and training concepts of target concepts (Mellish 1991). The process of identification can be considered as a search in the description spaces of L_d based on a partially ordered relation called "subsumption" ($\geq$). Informally, the relation states that description d_1 subsumes description d_2 ($d_1 \geq d_2$) if every element in L_d described by d_2 is also described by d_1. The latter definition allows to formalise the description identification task given below.

Description Identification Task $<L_d, <I^+, I^-, C^+, C^->>$

Given:
(1) A partially ordered description language L_d.
(2) Training sets I^+ and I^- of positive and negative instances of a target concept.
(3) Training sets C^+ and C^- of positive and negative concepts of a target concept.
Find: Consistent descriptions d of the target concept within L_d such that:

$$(\forall i \in I^+)(i \leq d) \wedge (\forall i \in I^-)\neg(i \leq d) \wedge$$
$$(\forall c \in C^+)(d \leq c) \wedge (\forall c \in C^-)\neg(d \leq c)$$

Therefore, description d is consistent with training data if (1) d does (not) subsume the positive (negative) instances i; and (2) d is (not) subsumed by the positive (negative) concepts c. The set of all consistent descriptions in L_d is known as the version space (*VS*) as it contains all possible descriptions (versions) of the target concept (Mitchell 1997). Thus, the identification can be considered as an incremental process of updating *VS* so that descriptions that classify the training data incorrectly are removed. The process completes when *VS* contains only one description which then is the description of the target concept.

There exist two basic algorithms for descriprtion identification. The first one is the description identification algorithm (DI) (Mellish 1991). It identifies version spaces to be learned when they are represented with their boundary sets *S* and *G* containing their most minimal and the most maximal descriptions. That is why the algorithm converges to the target description when the description is completely determined by the training data ($S = G$), but it is intractable as the boundary sets grow exponentially in the number of training elements in the worst case (Haussler 1988). In order to ovecome this computational problem the iterative version space algorithm (ITVS) has been developed in (Sablon, DeRaedt and Bruynooghe 1994; Sablon 1995). ITVS performs bi-directional depth search on the boundary sets of the version spaces to be learned and that is why it guarantees the polynomial complexity of the description identification process.

The version space representation methods, used by both algorithms, are restricted to the description identification task defined above. Therefore, when there exist dynamical changes of the classification of some of the training descriptions used, the methods cannot be incrementally updated so that version spaces are correctly represented w.r.t. changed training sets. The main reason for this negative result is the impossibility to determine the influence of training instances and concepts over version spaces when at least one boundary of the spaces is presented with a standard boundary set; i.e., training elements cannot be retracted.

This paper shows how to overcome this problem by introducing *integrated instance/concept-based boundary sets* (Smirnov and Braspenning 1998a; Smirnov and Braspenning 1998b) as a new version space representation method. The method determines two algorithms for description identification and retracting data that can be applied for the class of admissible languages. The algorithms are based on incremental version space merging (Hirsh 1990; Hirsh 1994) as intersecting is emulated with the "and" operator applied to the representations of the version spaces to be

intersected. That is why the algorithms are tractable when the boundary sets of a version space w.r.t. a training instance or concept can be computed in space and time polynomial in the relevant properties of the used languages.

Terminology, Definitions and Notation

Description language L_d is a set of descriptions. The descriptions are structured in L_d according to the partially ordered relation "subsumption". Structuring is used for organising a search for descriptions that satisfy constraints of description identification tasks. Characterising the search presupposes the following definitions from the theory of partially ordered sets to be used (Mitchell 1978).

Definition 1 If C is a poset then:

$\mathrm{MIN}(C) = \{ c \in C \mid (\forall c' \in C) \neg(c' < c) \}$
$\mathrm{MAX}(C) = \{ c \in C \mid (\forall c' \in C) \neg(c' > c) \}$

Definition 2 A poset C is a chain iff

$(\forall c_1, c_2 \in C)((c_1 \geq c_2) \vee (c_2 \geq c_1))$

Definitions 1 and 2 make it possible to determine when L_d is admissible for considering description identification as a search task (Mitchell 1978).

Definition 3 A description language L_d is admissible iff (1) L_d is a poset; and (2) every chain within L_d has just one minimal and just one maximal elements.

Concept C is represented in L_d with its corresponding description c. If c is not determined then C is represented with its version space based on its training instances and concepts.

Definition 4 (Mellish 1991) Version space VS w.r.t. a task $\langle L_d, \langle I^+, I^-, C^+, C^- \rangle\rangle$ is a set of consistent descriptions in L_d:

$VS = \{d \in L_d \mid \mathrm{cons}(d, \langle I^+, I, C^+, C^- \rangle)\}$

where cons is the consistent predicate:

$$\mathrm{cons}(d, \langle I^+, I, C^+, C^- \rangle) \leftrightarrow ((\forall i \in I^+)(i \leq d) \wedge (\forall i \in I)\neg(i \leq d) \wedge (\forall c \in C^+)(d \leq c) \wedge (\forall c \in C)\neg(d \leq c))$$

Definition 5 Version space VS w.r.t. a task $\langle L_d, \langle I^+, I^-, C^+, C^- \rangle\rangle$ is represented by the boundary sets S and G defined as follows:

$S = \mathrm{MIN}(VS)$ and $G = \mathrm{MAX}(VS)$.

Theorem 1 (Mellish 1991) **(Validity of the boundary sets)** Consider a task $\langle L_d, \langle I^+, I^-, C^+, C^- \rangle\rangle$ with version space VS and ordered pair of boundary sets $\langle S, G \rangle$. If L_d is admissible then:

$(\forall d \in L_d)((d \in VS) \leftrightarrow (\exists s \in S)(\exists g \in G)((s \leq d) \wedge (d \leq g)))$

Incremental Version Space Merging

Description identification can be considered as a process of intersection of version spaces based on different training data. This idea is employed by the incremental version space merging algorithm (IVSM) that emulates description identification according to theorem 2 (Hirsh 1990; Hirsh 1994).

Theorem 2 Consider a task $\langle L_d, \langle I^+_1, I^-_1, C^+_1, C^-_1 \rangle\rangle$ with version space VS_1, a task $\langle L_d, \langle I^+_2, I^-_2, C^+_2, C^-_2 \rangle\rangle$ with version space VS_2 and a task $\langle L_d, \langle I^+_1 \cup I^+_2, I^-_1 \cup I^-_2, C^+_1 \cup C^+_2, C^-_1 \cup C^-_2 \rangle\rangle$ with version space VS_{12}. Then: $VS_{12} = VS_1 \cap VS_2$.

The IVMS *emulation* of description identification starts with initialisation of the version space VS consistent with the first training element. After that for every new training element e its corresponding version space VS_e is formed. The formation of tne version space VS_e is followed by its intersection with the current version space VS. The resulting intersected space according to theorem 2 represents a new *updated* version space VS' of the target concept.
A complexity analysis shows that if version spaces are represented by their boundary sets and the intersection operators from (Mitchell 1978; Hirsh 1990; Hirsh 1994) are applied then the emulation is exponential in the worst case. This is due to the sets G and S that are *explicitly* built by *sequential assimilation* of the training elements; i.e., the time complexity of processing the n-th training description depends on the space complexity of the description identification process of the previous $n-1$ training descriptions.

Instance/Concept-Based Boundary Sets

The integrated instance/concept-based boundary sets of version spaces are developed in order to prevent the exponential growth of the sets S and G. The main idea is to represent the version space of the target concept as an intersection of simple version spaces corresponding to particular training elements (*lemma 1*) and to emulate the intersection with the "and" operator applied on their boundary sets (*definition 6*). That is why the new representation can be applied for admissible languages since according to theorem 1 every set of consistent descriptions in these languages is boundary sets representable. Formally, the scheme is stated in definition 1 preceded by lemma 1.

Lemma 1 If VS is the version space of a task $\langle L_d, \langle I^+, I^-, C^+, C^- \rangle\rangle$ then:

$$VS = \bigcap_{p=1}^{|I^+||C^-|} \bigcap_{n=1} VS_{p,n} = \bigcap_{p=1}^{|I^+||C^-|} \bigcap_{n=1} VS_{p,n} \quad \text{where}$$

$VS_{p,n} = \{d \in L_d \mid \mathrm{cons}(d, \langle i_p, I, C^+, c_n \rangle)\}$;

$VS_{n,p} = \{ d \in L_d \mid \mathrm{cons}(d, \langle I^+, i_n, c_p, C^- \rangle)\}$.

Proof. The lemma is a corollary of theorem 2.

Definition 6 (Integrated instance/concept-based boundary sets) Version space VS of a task $\langle L_d, \langle I^+, I^-, C^+, C^- \rangle\rangle$ is represented with a ordered pair of two sequences of boundary sets $\langle\langle S_{1,1}, \ldots, S_{Pi,Nc}\rangle, \langle G_{1,1}, \ldots, G_{Ni,Pc}\rangle\rangle$ defined as follows:

$S_{p,n} = \mathrm{MIN}(VS_{p,n})$ for $p \in [1,Pi]$ and $n \in [1,Nc]$;
$G_{n,p} = \mathrm{MAX}(S_{n,p})$ for $n \in [1,Ni]$ and $p \in [1,Pc]$;

where $Pi = |I^+|$, $Nc = |C^-|$, $Ni = |I^-|$ and $Pc = |C^+|$.

The integrated instance/concept-based boundary sets get their name from the fact that the boundary sets $S_{p,n}$ and $G_{n,p}$ correspond to simple version spaces $VS_{p,n}$ and $VS_{n,p}$ respectively that are associated with unique training elements i_p, i_n, c_p and c_n. Every one disjunction of the elements of the minimal (maximal) sets $S_{p,n}$ and $G_{n,p}$ determines eventual belonging to $VS_{p,n}$ and $VS_{n,p}$ so that the *conjunction of these disjunctions* determines belonging to version space *VS* (which is proven in theorem 3). Therefore, the real maximal and minimal boundary sets *S* and *G* of version space *VS* are *implicitly* expressed in *CNF* extended description languages.

Theorem 3 (Validity of the integrated instance/concept-based boundary sets) Consider a task $<L_d,<I^+, I^-, C^+, C^->>$ with version space *VS* and instance/concept-based boundary sets $<<S_{1,1},\ldots, S_{Pi,Nc}>, <G_{1,1},\ldots, G_{Ni,Pc}>>$. Therefore, if language L_d is admissible then:

$(\forall d\in L_d)((d\in VS) \leftrightarrow$
$(\exists s_{1,1},..,s_{Pi,Nc}\in S_{1,1},..,S_{Pi,Nc})(\exists g_{1,1},..,g_{Ni,Pc}\in G_{1,1},..,G_{Ni,Pc})$
$((s_{1,1}\le d)...\wedge(\ s_{p,n1}\le d)\wedge(d\le g_{1,1})...\wedge(d\le g_{p,n})))$

Proof. ($\leftarrow$) Consider arbitrarily chosen d ($d \in L_d$) and a sequence $s_{1,1},\ldots, s_{Pi,Nc}$ ($s_{1,1},\ldots, s_{Pi,Nc}\in S_{1,1},\ldots, S_{Pi,Nc}$) so that (1) ($s_{1,1}\le d$) ...$\wedge$ ($s_{Pi,Nc}\le d$). As $s_{p,n}$ ($s_{p,n}\in S_{p,n}$) belongs to its corresponding space $VS_{p,n}$ then according to the definition of $VS_{p,n}$ follows that cons($s_{p,n}$, $<i_p, I, C^+, c_n>$). Therefore, (2) $i_p\le s_{p,n}$ and (3) $\neg(\ s_{p,n}\le c_n)$ for $p\in[1, Pi]$) and $n\in[1, Nc]$.

From (1) and (2) follows that $(i_1\le s_{1,1}\le d)...\wedge(\ i_{Pi}\le s_{Pi,Nc}\le d)$ that leads to $(i_1\le d)...\wedge(\ i_{Pi}\le d)$ because relation "$\le$" is a partial ordering. The last derivation can be re-written as (4) $(\forall i\in I^+)(i\le d)$ because the instances $i_1 \ldots, i_{Pi}$ form the set I^+.

From (1) and (3) follows that $(s_{1,1}\le d)\wedge\neg(\ s_{1,1}\le c_1)...\wedge(s_{Pi,Nc}\le d)\wedge\neg(\ s_{Pi,Nc}\le c_{Nc})$ that leads to $\neg(d\le c_1)...\wedge\neg(d\le c_{Nc})$ as relation "$\le$" is a partial ordering. The last derivation can be re-written as (5) $(\forall c\in C^-)\neg(d\le c)$ because the training concepts $c_1 \ldots, c_{Nc}$ form the set C^-.

Consider the same d ($d\in L_d$) and arbitrarily chosen sequence $g_{1,1},\ldots, g_{Ni,Pc}$ ($g_{1,1},\ldots, g_{Ni,Pc}\in G_{1,1},\ldots,G_{Ni,Pc}$) so that (6) $(d\le g_{1,1})...\wedge(d\le g_{Ni,Pc})$. As $g_{n,p}$ ($g_{n,p}\in G_{n,p}$) belongs to its corresponding $VS_{n,p}$ then according to the definition of $VS_{n,p}$ follows that cons($g_{n,p}$, $<I^+, i_n, c_p, C^->$). Therefore, (7) $\neg(i_n\le g_{n,p})$ and (8) $(g_{n,p}\le c_p)$ for all $n\in[1, Ni]$ and $p\in[1, Pc]$.

From (6) and (7) follows that $(d\le g_{1,1})\wedge\neg(i_1\le g_{1,1})...\wedge(d\le g_{Ni,Pc})\wedge\neg(i_{Ni}\le g_{Ni,Pc})$ that leads to $\neg(i_1\le d)...\wedge\neg(\ i_{Ni}\le d)$ because relation "$\le$" is a partial ordering. The last derivation can be re-written as (8) $(\forall i\in I)\neg(i\le d)$ because the training instances $i_1 \ldots, i_{Ni}$ form the set I.

From (6) and (8) follows that $(d\le g_{1,1}\le c_1)...\wedge(d\le g_{n,p}\le c_{Pc})$ that leads to $(d\le c_1)...\wedge(d\le c_{Pc})$ as relation "$\le$" is a partial ordering. The last derivation can be re-written as (9) $(\forall c\in C^+)(d\le c)$ because the training concepts $c_1 \ldots, c_{Pc}$ form the set C^+.

By (4), (5), (8) and (9) follows that cons(d, $<I^+, I, C^+, C^->$) according to the definition of the consistency predicate; i.e, $d\in VS$ and the first part of the theorem is proven.

($\rightarrow$) Consider arbitrarily chosen d such that $d\in VS$. According to lemma 1 d belongs simultaneously to version spaces $VS_{p,n}$ and $VS_{n,p}$. Therefore, according to theorem 1 as L_d is admissible the following two derivations hold:

(1) $(d\in VS)\rightarrow(\exists s_{1,1},\ldots,s_{Pi,Nc}\in S_{1,1},\ldots, S_{Pi,Nc})((s_{1,1}\le d)...\wedge(s_{p,n1}\le d))$
(2) $(d\in VS)\rightarrow(\exists g_{1,1},\ldots,g_{Ni,Pc}\in G_{1,1},\ldots,G_{Ni,Pc})((d\le g_{1,1})...\wedge(d\le g_{p,n}))$

By (1) and (2)

$(d\in VS)\rightarrow$
$(\exists s_{1,1},\ldots,s_{Pi,Nc}\in S_{1,1},\ldots, S_{Pi,Nc})(\exists g_{1,1},\ldots,g_{Ni,Pc}\in G_{1,1},\ldots,G_{Ni,Pc})$
$((s_{1,1}\le d)...\wedge(\ s_{p,n1}\le d)\wedge(d\le g_{1,1})...\wedge(d\le g_{p,n}))$

and the second part of the theorem is proven.

Description Identification Algorithm

The proposed representation specialises theorem 2 to theorem 4 for the cases of identification with training positive and negative instances and concepts. The theorem determines an incremental algorithm of description identification which is an extension the IVMS algorithm. The algorithm is presented in figure 1.

Theorem 4 Consider a task $<L_d,<I^+, I^-, C^+, C^->>$ with version space *VS* and instance/concept-based boundary sets $<<S_{1,1},\ldots,S_{Pi,Nc}>,<G_{1,1},\ldots,G_{Ni,Pc}>>$.

1. If $<L_d, <I^+', I^-, C^+, C^->>$ is a new task with version space *VS'* and instance/concept-based boundary sets $<<S'_{1,1},\ldots, S'_{Pi+1,Nc}>, <G'_{1,1},\ldots, G'_{Ni,Pc}>>$ s.t. $I^{+\prime} = I^+\cup\{i_{Pi+1}\}$ then:
For $n\in[1, Ni]$ and $p\in[1, Pc]$
$G'_{n,p} = \{\ g\in G_{n,p}\mid(\ i_{Pi+1}\le g)\ \}$;
For $p\in[1, Pi]$ and $n\in[1, Nc]$
$S'_{p,n} = S_{p,n}$;
For $n\in[1, Nc]$
$S'_{Pi+1,n}$= MIN($\{d\in L_d\mid$ cons(d,$<i_{Pi+1}, I^-, C^+, c_n>$)$\}$);

2. If $<L_d, <I^+, I^-', C^+, C^->>$ is a new task with version space *VS'* and instance/concept-based boundary sets $<<S'_{1,1},\ldots, S'_{Pi,Nc}>, <G'_{1,1},\ldots, G'_{Ni+1,Pc}>>$ s.t. $I^{-\prime} = I^-\cup\{i_{Ni+1}\}$ then:
For $p\in[1, Pi]$ and $n\in[1, Nc]$
$S'_{p,n} = \{\ s\in S_{p,n}\mid\neg(i_{Ni+1}\le s)\ \}$;
For $n\in[1, Ni]$ and $p\in[1, Pc]$
$G'_{n,p} = G_{n,p}$;
For $p\in[1, Pc]$
$G'_{Ni+1,p}$= MAX($\{d\in L_d\mid$ cons(d,$<I^+, i_{Ni+1}, c_p, C^->$)$\}$);

3. If $<L_d, <I^+, I^-, C^+', C^->>$ is a new task with version space *VS'* and instance/concept-based boundary sets $<<S'_{1,1},\ldots, S'_{Pi,Nc}>, <G'_{1,1},\ldots,G'_{Ni,Pc+1}>>$ s.t. $C^{+\prime} = C^+\cup\{c_{Pc+1}\}$ then:
For $p\in[1, Pi]$ and $n\in[1, Nc]$
$S'_{p,n} = \{\ s\in S_{p,n}\mid(s\le c_{Pc+1})\ \}$;
For $n\in[1, Ni]$ and $p\in[1, Pc]$
$G'_{n,p} = G_{n,p}$;
For $n\in[1, Ni]$
$G'_{n, Pc+1}$= MAX($\{d\in L_d\mid$ cons(d,$<I^+, i_n, c_{Pc+1}, C^->$)$\}$);

4. If $<L_d, <I^+, I^-, C^+, C^-'>>$ is a new task with version space *VS'* and instance/concept-based boundary sets $<<S'_{1,1},\ldots, S'_{P,Nc+1}>, <G'_{1,1},\ldots, G'_{Ni,Pc}>>$ s.t. $C^{-\prime} = C^-\cup\{c_{Nc+1}\}$ then:

For all $n \in [1, Ni]$ and $p \in [1, Pc]$
$G'_{n,p} = \{ g \in G_{n,p} \mid \neg(c_{Nc+1} \leq g) \}$;
For $p \in [1, Pi]$ and $n \in [1, Nc]$
$S'_{p,n} = S_{p,n}$;
For $p \in [1, Pi]$
$S'_{p, Nc+1} = \text{MIN}(\{d \in L_d \mid \text{cons}(d, <i_p, I, C^+, c_{Nc+1}>)\})$;

Proof. *(Proof of the first part of the theorem)* As the new updated version space *VS'* can be considered as intersection of version spaces $VS'_{p,n}$ and $VS'_{n,p}$ according to lemma 1, then the proof of the theorem is accomplished by determining their boundary sets.

A). Version spaces of $VS'_{n,p}$ are changed for all $n \in [1, Ni]$ and for all $p \in [1, Pc]$ as

$$VS'_{n,p} = \{d \in L_d \mid \text{cons}(d, <I^+', i_n, c_p, C^->)\} = \{ d \in L_d \mid \text{cons}(d, <I^+ \cup \{i_{Pi+1}\}, i_n, c_p, C^->)\}$$

according to lemma 1. Therefore,

$$G'_{n,p} = \{ g \in G_{n,p} \mid (i_{Pi+1} \leq g) \}$$

for all $n \in [1, Ni]$ and for all $p \in [1, Pc]$ according to the theorem for revising boundary sets given a positive example (Mellish, 1991).

B). Version spaces $VS'_{p,n}$ are not changed for all $p \in [1, Pi]$ and for all $n \in [1, Nc]$ as

$VS'_{p,n} = VS_{p,n} = \{ d \in L_d \mid \text{cons}(d, < i_p, I^-, C^+, c_n>)\}$ according to lemma 1.

At the same time instance i_{Pi+1} leads to inducing new version spaces $VS'_{Pi+1,n}$ for $n \in [1, Nc]$:

$$VS'_{Pi+1,n} = \{ d \in L_d \mid \text{cons}(d, < i_{Pi+1}, I^-, C^+, c_n>)\}$$

according to lemma 1. Therefore,

$$S'_{Pi+1,n} = \text{MIN}(VS'_{Pi+1,n}) = \text{MIN}(\{ d \in L_d \mid \text{cons}(d, < i_{Pi+1}, I^-, C^+, c_n>)\})$$

according to definition 5.

The proof of parts (2), (3) and (4) of the theorem can be derived by analogy and duality.

The algorithm initiates the description identification process by initialising the version space similar to IVMS (figure 1). After initialisation it learns *incrementally* the version space of the target concept. Upon entry of a positive instance i_{Pi+1} it removes all elements of the maximal boundary sets $G_{n,p}$ that do *not* subsume the instance. This guarantees that the updated sets $G'_{n,p}$ are the maximal boundary sets of the spaces $VS'_{n,p}$. The minimal boundary sets $S_{p,n}$ of the version spaces $VS_{p,n}$ remain unchanged according to theorem 4, and the minimal sets $S'_{Pi+1,n}$ are initialised as sets of minimal elements in L_d that are consistent with the sets I^- and C^+, the instance i_{Pi+1}, and corresponding negative training concepts c_n for all $n \in [1, Nc]$.
Upon entry of a negative instance i_{Ni+1} the algorithm removes all elements of minimal boundary sets $S_{p,n}$ that do subsume the instance. This guarantees that the updated sets $S'_{p,n}$ are the minimal boundary sets of the spaces $VS'_{p,n}$. The

IF current training element *e* is a positive instance i_{Pi+1}
FOR $n = 1$ to $|I|$ **DO**
FOR $p = 1$ to $|C^+|$ **DO**
$G'_{n,p} = \{ g \in G_{n,p} \mid (i_{Pi+1} \leq g) \}$;
FOR $n = 1$ to $|C^-|$ **DO**
$S'_{Pi+1,n} = \text{MIN}(\{ d \in L_d \mid \text{cons}(d, < i_{Pi+1}, I^-, C^+, c_n>)\})$;

IF current training element *e* is a negative instance i_{Ni+1}
FOR $p = 1$ to $|I^+|$ **DO**
FOR $n = 1$ to $|C^-|$ **DO**
$S'_{p,n} = \{ s \in S'_{p,n} \mid \neg(i_{Ni+1} \leq s) \}$;
FOR $p = 1$ to $|C^+|$ **DO**
$G'_{Ni+1,p} = \text{MAX}(\{ d \in L_d \mid \text{cons}(d, <I^+, i_{Ni+1}, c_p, C^->)\})$;

IF current training element *e* is a positive concept c_{Pc+}
FOR $p = 1$ to $|I^+|$ **DO**
FOR $n = 1$ to $|C|$ **DO**
$S'_{p,n} = \{ s \in S_{p,n} \mid (s \leq c_{Pc+1}) \}$;
FOR $n = 1$ to $|I|$ **DO**
$G'_{n,Pc+1} = \text{M AX}(\{ d \in L_d \mid \text{cons}(d, <I^+, i_n, c_{Pc+1}, C^->)\})$;

IF current training element *e* is a negative concept c_{Nc+1}
FOR $n = 1$ to $|I|$ **DO**
FOR $p = 1$ to $|C^+|$ **DO**
$G'_{n,p} = \{ g \in G_{n,p} \mid \neg(c_{Nc+1} \leq g) \}$;
FOR $p = 1$ to $|I^+|$ **DO**
$S'_{p, Nc+1} = \text{MIN}(\{d \in L_d \mid \text{cons}(d, <i_p, I, C^+, c_{Nc+1}>)\})$

Figure 1. The Description Identification Algorithm of the Integrated Instance/Concept-Based Boundary Sets

maximal boundary sets $G_{n,p}$ remain unchanged according to theorem 4 and for every c_p ($p \in [1,Pc]$) its corresponding maximal set $G_{Ni+1,p}$ is generated as a set of all maximal descriptions in L_d, that are consistent with the sets I^+ and C^-, the instance i_{Ni+1} and the training concept c_p.
The behavior of the algorithm for positive and negative concepts is dual to the ones of positive and negative instances; hence it is analogous in form.

Retraction Algorithm

The instance/concept-based boundary sets are suitable for retracting used training descriptions because the influence of instances and concepts over version spaces can be identified. This follows from definition 6 which states that the boundaries of the target version space is derived by conjunctive linking boundary sets of simple version spaces. Therefore, if the version space boundary set corresponding to a training description can be found and after that removed then the resulting target version space is revised to a new one that is *not* based on that description.
Below the correctness (theorem 5) of the retraction algorithm for the instance/concept-based boundary sets is shown for the cases of positive and negative instances and concepts. Theorem 5 is based on lemma 2. The retraction algorithm is presented in figure 2.

Lemma 2 Consider tasks $<L_d, < I^+, I, C^+, C^->>$, $<L_d, <I^+ \cup \{e\}, I, C^+, C^->>$ and $<L_d, <I^+, I \cup \{e\}, C^+, C^->>$ with version spaces VS_1, VS_2 and VS_3. Then:

$VS_1 = VS_2 \cup VS_3$.

Theorem 5 Consider a task $<L_d, <I^+, I^-, C^+, C^->>$ with version space VS and instance/ concept-based boundary sets $<<S_{1,1},\ldots, S_{Pi,Nc}>, <G_{1,1},\ldots, G_{Ni,Pc}>>$.

1. If $<L_d, <I^+', I^-, C^+, C^->>$ is a new task with version space VS' and instance/concept-based boundary sets $<<S'_{1,1},\ldots, S'_{Pi-1,Nc}>, <G'_{1,1},\ldots, G'_{Ni,Pc}>>$ s.t. $I^+' = I^+ - \{i_{Pi}\}$ then:

Generate:

$Gi_{Pi} = \text{MAX}(\{d \in L_d \mid \text{cons}(d, <I^+ - \{i_{Pi}\}, \{i_n\} \cup \{i_{Pi}\}, c_p, C^->)\})$;

For $n \in [1, Ni]$ and $p \in [1, Pc]$

$G'_{n,p} = \text{MAX}(G_{n,p} \cup Gi_{Pi})$;

For $n \in [1, Nc]$ the sets $S_{Pi,n}$ are removed.

2. If $<L_d, <I^+, I^-', C^+, C^->>$ is a new task with version space VS' and instance/concept-based boundary sets $<<S'_{1,1},\ldots, S'_{Pi,Nc}>, <G'_{1,1},\ldots, G'_{Ni-1,Pc}>>$ s.t. $I^-' = I^- - \{i_{Ni}\}$ then:

Generate:

$Si_{Ni} = \text{MIN}(\{d \in L_d \mid \text{cons}(d, <\{i_p\} \cup \{i_{Ni}\}, I^- - \{i_{Ni}\}, C^+, c_n>)\})$;

For $p \in [1, Pi]$ and $n \in [1, Nc]$

$S'_{p,n} = \text{MIN}(S_{p,n} \cup Si_{Ni})$;

For $p \in [1, Pc]$ the sets $G_{Ni,p}$ are removed.

3. If $<L_d, < I^+, I^-, C^+', C^->>$ is a new task with version space VS' and instance/concept- based boundary sets $<<S'_{1,1},\ldots, S'_{Pi,Nc}>, <G'_{1,1},\ldots, G'_{Ni,Pc-1}>>$ s.t. $C^+' = C^+ - \{c_{Pc}\}$ then:

Generate:

$Sc_{Pc} = \text{MIN}(\{d \in L_d \mid \text{cons}(d, <i_p, I, C^+ - \{c_{Pc}\}, \{c_n\} \cup \{c_{Pc}\}>)\})$;

For $p \in [1, Pi]$ and $n \in [1, Nc]$

$S'_{p,n} = \text{MIN}(S_{n,p} \cup Sc_{Pc})$;

For all $n \in [1, Ni]$ the sets $G_{n,Pc}$ are removed.

4. If $<L_d, <I^+, I^-, C^+, C^-'>>$ is a new task with version space VS' and instance/concept-based bo-undary sets $<<S'_{1,1},\ldots, S'_{P,Nc-1}>, <G'_{1,1},\ldots, G'_{Ni,Pc}>>$ s.t. $C^-' = C^- - \{c_{Nc}\}$ then:

Generate:

$Gc_{Nc} = \text{MAX}(\{d \in L_d \mid \text{cons}(d, <I^+, \{i_n\}, \{c_p\} \cup \{c_{Nc}\}, C^- - \{c_{Nc}\}>)\})$;

For $n \in [1, Ni]$ and $p \in [1, Pc]$

$G'_{n,p} = \text{MAX}(G_{n,p} \cup Gc_{Nc})$;

For all $p \in [1, Pc]$ the sets $S_{p,Nc}$ are removed.

Proof. *(Proof of the fist part of the theorem)* As the new updated version space VS' can be considered as intersection of version spaces $VS'_{p,n}$ and $VS'_{n,p}$ according to lemma 1, then the proof of the theorem is accomplished by determining their boundary sets.

A). Version spaces of $VS'_{n,p}$ are changed for all $n \in [1, Ni]$ and for all $p \in [1, Pc]$ as

$$VS'_{n,p} = \{d \in L_d \mid \text{cons}(d, <I^+', i_n, c_p, C^->)\} = \{ d \in L_d \mid \text{cons}(d, <I^+ - \{i_{Pi}\}, i_n, c_p, C^->)\}$$

according to lemma 1. Therefore, according to lemma 2 follows that:

$$VS'_{n,p} = \{ d \in L_d \mid \text{cons}(d, <I^+, i_n, c_p, C^->)\} \cup \{ d \in L_d \mid \text{cons}(d, <I^+ - \{i_{Pi}\}, \{i_n\} \cup \{i_{Pi}\}, c_p, C^->)\}$$

But $\{d \in L_d \mid \text{cons}(d, <I^+, i_n, c_p, C^->)\}$ is the definition of the version spaces $VS_{n,p}$. Therefore,

$$VS'_{n,p} = VS_{n,p} \cup \{ d \in L_d \mid \text{cons}(d, <I^+, \{i_n\} \cup \{i_{Pi}\}, c_p, C^->)\}$$

Thus, $G'_{n,p} = \text{MAX}(VS'_{n,p})$

$$G'_{n,p} = \text{MAX}(\text{MAX}(VS_{n,p}) \cup \text{MAX}(\{d \in L_d \mid \text{cons}(d, <I^+ - \{i_{Pi}\}, \{i_n\} \cup \{i_{Pi}\}, c_p, C^->)\}))$$

If the set $\text{MAX}(\{d \in L_d \mid \text{cons}(d, <I^+ - \{i_{Pi}\}, \{i_n\} \cup \{i_{Pi}\}, c_p, C^->)\})$ is denoted as a set Gi_{Pi} then:

$G'_{n,p} = \text{MAX}(G_{n,p} \cup Gi_{Pi})$

B). Version spaces $VS'_{p,n}$ are not changed for all $p \in [1, Pi-1]$ and for all $n \in [1, Nc]$ as

$$VS'_{p,n} = VS_{p,n} = \{ d \in L_d \mid \text{cons}(d, < i_p, I^-, C^+, c_n>)\}$$

The influence of the instance i_{Pi} on the version space VS' is removed if $S_{Pi,n}$ is removed for all $n \in [1, Nc]$ from the instance/concept-based boundary sets of VS' which follows from lemma 1.

The proof of parts (2), (3) and (4) of the theorem can be derived by analogy and duality.

IF the element e to be retracted is a positive instance i_{Pi}
$Gi_{Pi} = \text{MAX}(\{d \in L_d \mid \text{cons}(d, <I^+ - \{i_{Pi}\}, \{i_n\} \cup \{i_{Pi}\}, c_p, C^->)\})$;
FOR $n = 1$ to $|I|$ **DO**
  FOR $p = 1$ to $|C^+|$ **DO**
    $G'_{n,p} = \text{MAX}(G_{n,p} \cup Gi_{Pi})$;
FOR $n = 1$ to $|C|$ **DO**
  $S_{Pi,n}$ is removed;

IF the element e to be retracted is a negative instance i_{Ni}
$Si_{Ni} = \text{MIN}(\{d \in L_d \mid \text{cons}(d, <\{i_p\} \cup \{i_{Ni}\}, I - \{i_{Ni}\}, C^+, c_n>)\})$;
FOR $p = 1$ to $|I^+|$ **DO**
  FOR $n = 1$ to $|C|$ **DO**
    $S'_{p,n} = \text{MIN}(S_{p,n} \cup Si_{Ni})$;
FOR $p = 1$ to $|C^+|$ **DO**
  $G_{Ni,p}$ is removed;

IF the element e to be retracted is a positive concept c_{Pc}
$Sc_{Pc} = \text{MIN}(\{d \in L_d \mid \text{cons}(d, <i_p, I, C^+ - \{c_{Pc}\}, c_n \cup \{c_{Pc}\}>)\})$;
FOR $p = 1$ to $|I^+|$ **DO**
  FOR $n = 1$ to $|C|$ **DO**
    $S'_{p,n} = \text{MIN}(S_{n,p} \cup Sc_{Pc})$;
FOR $p = 1$ to $|C^-|$ **DO**
  $G_{n,Pc+1}$ is removed;

IF the element e to be retracted is a negative concept c_{Nc}
$Gc_{Nc} = \text{MAX}(\{d \in L_d \mid \text{cons}(d, <I^+, \{i_n\}, c_p \cup \{c_{Nc}\}, C^- - \{c_{Nc}\}>)\})$;
FOR $n = 1$ to $|I|$ **DO**
  FOR $p = 1$ to $|C^+|$ **DO**
    $G'_{n,p} = \text{MAX}(G_{n,p} \cup Gc_{Nc})$;
FOR $p = 1$ to $|I^+|$ **DO**
  $S_{p,Nc-1}$ is removed;

Figure 2. The Retraction Algorithm of the Integrated Instance/Concept Based Boundary Sets

Applying the retraction algorithm presupposes that the version space has been learned by a non-empty set of training data (figure 2). When a positive instance i_{Pi} has to be retracted then each updated set $G'_{n,p}$ is formed as the maximum of the union of the previous set $G_{n,p}$ and the set Gi_P =MAX({ $d \in L_d$ | cons(d, <I^+ - {i_{Pi}}, {i_n}$\cup${i_{Pi}}, c_p, C^->)}). This guarantees that the sets $G'_{n,p}$ are the maximal sets of the updated version spaces $VS'_{n,p}$; i.e., the influence of the instance over the version spaces $VS'_{n,p}$ is removed. In order to complete the operation, the minimal boundary sets $S_{Pi,n,}$ of the version spaces $VS_{Pi,n}$ associated with the instance i_{Pi} , are removed from the list of the *S*-sets for all $n \in [1, Nc]$. This means that the instance does not already determine the boundaries of the entire version space; i.e., the target version space is restored from the instance.

When a negative instance i_{Ni} has to retracted then each updated set $S'_{p,n}$ is formed as the minimum of the union of the previous set $S_{p,n}$ and the set Si_{Ni}= MIN({$d \in L_d$ | cons(d,<{i_p}$\cup${ i_{Ni}}, I^- - {i_{Ni}}, C^+, c_n>)}). This guarantees that the sets $S'_{p,n}$ are the maximal sets of the updated version spaces $VS'_{p,n}$; i.e., the influence of the instance over the version spaces $VS'_{p,n}$ is removed. In order to complete the operation, the maximal boundary sets $G'_{Ni,p}$, associated with the instance i_{Ni}, are removed from the list of the G-sets for all $p \in [1, Pc]$. This means that the instance does not already determine the boundaries of the version space; i.e., the version space is completely restored from the instance.

The behavior of the algorithm for positive and negative concepts is dual to the ones of positive and negative instances; hence it is analogous in form.

Complexity Analysis

The complexity analysis of the description identification and the retraction algorithms is made in terms of *Pi*, *Ni*, *Pc* and *Nc* respectively the numbers of positive and negative instances and concepts; and Σ and Γ respectively the largest sizes of boundary sets *S* and *G* of a simple version space. Thus, the time complexity is determined as a sum of the boundaries of the number of description's manipulations in L_d while the space complexity is determined as a quantitative bound of the space for saving concept descriptions.

Complexity of the Description Identification Algorithm

The time complexity of processing of one positive instance i_{Pi+1} is equal to the sum of the complexities of updating sets $G_{n,p}$ plus the complexity of deriving new set $S'_{Pi+1,n}$. The complexity of updating set $G_{n,p}$ is O($Ni\ Pc\ \Gamma$); and the complexity of deriving sets $S'_{Pi+1,n}$ is O(Nc G($S'_{Pi+1,n}$)) where G($S'_{Pi+1,n}$) is the time complexity for generating one set $S'_{Pi+1,n}$. Therefore, the overall complexity of processing a positive instance is O($Ni\ Pc\ \Gamma$ + Nc G($S'_{Pi+1,n}$)).

The time complexity of processing of a negative instance is analogously derived and it is O($Pi\ Nc\ \Sigma + Pc$ G(G'_{Ni+1p})) where G(G'_{Ni+1p}) is complexity of generation of one set G'_{Ni+1p}.

The time complexities of processing positive and negative concepts are derived dually and they are O($Pi\ Nc\ \Sigma + Ni$ G($G'_{n,Pc+1}$)) and O($NiPc\ \Gamma + Pi$ G($S'_{p,Nc+1}$)) respectively, where G($G'_{n,Pc+1}$) and G($S'_{p,Nc+1}$) are time complexities for generating the sets $G'_{n,Pc+1}$ and $S'_{p,Nc+1}$.

Complexity of the Retraction Algorithm

The time complexity of processing of one positive instance i_{Pi} is equal to the sum of the complexities of updating the sets $G'_{n,p}$ plus the sum of the complexities of removing the sets S'_{Pip}. The first sum is equal to O(G(Gi_{Pi})+$Ni\ Pc\ \Gamma^2$) where G(Gi_{Pi}) is the complexity of generation of the set Gi_{Pi} and Γ^2 is the complexity of the operation MAX. The second sum is O(Nc) as Nc number of S_{Pin} sets are removed. Therefore, the overall complexity of processing of one positive instance i_{Pi} is O(G(Gi_{Pi}) +$Ni\ Pc\ \Gamma^2$).

The time complexity of processing of a negative instance i_{Ni} is analogously derived and it is O(G(Si_{Ni}) +$Pi\ Nc\ \Sigma^2$) where G(Si_{Ni}) is the complexity of generating the set Si_{Ni}.

The time complexities of processing positive and negative training concepts are derived dually and they are O(G(Sc_{Pc}) +$Pi\ Nc\ \Sigma^2$) and O(G(Gc_{Ni}) +$Ni\ Pc\Gamma^2$) respectively where G(Sc_{Pc}) and G(Gc_{Ni}) are the complexity of generation of the sets Sc_{Pc} and Gc_{Ni}.

The space complexity of both algorithms remains the same and it is equal to the space complexity of the integrated instance/concept-based boundary sets that is O($Pi\ Nc\ \Sigma + Ni\ Pc\Gamma$) where O($Pi\ Nc\ \Sigma$) and O($Ni\ Pc\ \Gamma$) correspond to the minimal and the maximal boundary sets $S_{p,n}$ and $G_{n,p}$ respectively.

Example

Consider a description identification task <L_d,<I^+= {[3, 3]}, I^-= {[5,5]}, C^+= {[2, 5]}, C^-= {[1, 3], [4, 5]}>> where L_d is the description language given in Figure 3.

Therefore, according to lemma 1 the boundary sets of simple version spaces are defined as follows $S_{[3,\ 3],\ [1,\ 3]}$ = {[3,4]}; $S_{[3,\ 3],\ [4,\ 5]}$ = {[3, 3]} and $G_{[5,5],\ [2,5]}$ = {[2,4]}. Thus, according to definition 6 the integrated instance/concept-based boundary sets of the version space to be learned is: << {[3,4]}, {[3, 3]}>, <{[2,4]}>>.

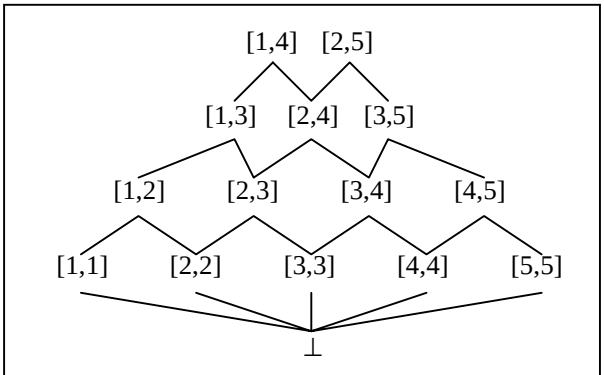

Figure 3. The description language of intervals

If the training concept [1, 3] has to be retracted then the retraction algorithm updates the set $G_{[5,5],[2,5]}$ as follows:

$G'_{[5,5],[2,5]} = \text{MAX}(G_{[5,5],[2,5]} \cup G_{[1,3]})$

But $G_{[5,5],[2,5]} = \{[2,4]\}$ and

$G_{[1,3]} = \text{MAX}(\{d \in L_d \mid \text{cons}(d, <[3,3], [5,5], [2,5] \cup [1,3], [4,5]>)\}) = \{[2,4]\}$

Therefore, $G'_{[5,5],[2,5]} = \{[2, 4]\}$

After that the retraction algorithm removes the minimal boundary sets $S_{[3,3],[1,3]}$ and the resulting integrated instance/ concept-based boundary sets of the version space to be learned becomes:

<< {[3, 3]} >, <{[2, 4]}>>.

Conclusion

This paper shows that the integrated instance/concept-based boundary sets is a new version space representation method that can be used for *description identification* and *retracting training data* for the whole class of admissible description languages. Moreover, it has been shown that the complexities of the algorithms of these two important operations depend on the complexities of generation and space complexities of boundary sets of simple version spaces. Therefore, it is possible to conclude that if the boundary sets of a simple version spaces can be computed in space and time polynomial in the relevant properties of the used languages then the integrated instance/concept-based boundary sets ensure the tractability of both algorithms.

Future research will *transfer* the integrated instance/concept-based boundary sets *to the description identification and retracting training data in the context of disjunctive version spaces* (Mitchell 1978; Sablon 1995; Sebag 1996; Sebag and Rouveirol 1997; Smirnov and Braspenning 1997). This will allow to circumvent the computational problems of this type of learning as well as the problem of identification in the presence of noisy training elements for a broad class of description languages.

References

Haussler, D. 1988. Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. *Artificial Intelligence* 36:177-221.

Hirsh, H. 1990. *Incremental Version Space Merging: A General Framework for Concept Learning.* Kluwer.

Hirsh, H. 1994. Generalizing Version Spaces. *Machine Learning* 17:5-46.

Mellish, C. 1991. The description Identification Problem. *Artificial Intelligence* 52: 151-167.

Mitchell, T. 1978. Version Space - An Approach to Concept Learning. Ph.D. diss., Stanford University.

Mitchell, T. 1997. *Machine Learning*, McGraw-Hill.

Sablon, G. 1995. Iterative Versionspaces with an Application in Inductive Logic Proramming. Ph.D. diss., Katholeke Universiteit Leuven, Belgium.

Sablon, G., De Raedt, L., and Bruynooghe, M. 1994. Iterative Versionspaces. *Artificial Intelligence*, 69: 393-409.

Sebag, M. 1996. Delaying the Choice of Bias: A Disjunctive Version Space Approach. In Proceedings of the Thirteenth International Conference on Machine Learning.

Sebag, M. and Rouveirol, C. 1997. Tractable Induction and Classification in First Order Logic via Stochastic Matching. In Proceedings of International Joint Conference on Artificial Intelligence , 888-893.

Smirnov, E. N., and Braspenning, P.J. 1997. A Space Fragmenting Family of Version Space Algorithms. In Proceedings of the Ninth Dutch Conference on Artificial Intelligence, (NAIC'97), 313-322.

Smirnov, E.N., and Braspenning, P.J. 1998a. Version Space Learning with Instance-Based Boundary Sets. In Proceedings of the Thirteenth European Conference on Artificial Intelligence, 460-464. John Wiley & Sons.

Smirnov, E.N., and Braspenning, P.J. 1998b. Description Identification with Integrated Instance-Based Boundary Sets. In Proceedings of The Eight Belgian-Dutch Conference on Machine Learning (BENELEARN'98), 62-70.