Lattice Machine: Version Space in Hyperrelations

[Extended Abstract] *

Hui Wang, Ivo Düntsch
School of Information and Software Engineering
University of Ulster
Newtownabbey, BT 37 0QB, N.Ireland
{H.Wang|I.Duentsch}@ulst.ac.uk

Günther Gediga
Institut für Evaluation und Marktanalysen
Brinkstr. 19
49143 Jeggen, Germany
gediga@eval-institut.de

Andrzej Skowron
Institute of Mathematics
University of Warsaw
02-097 Warszawa, Poland
skowron@mimuw.edu.pl

ABSTRACT

A version space is a set of all hypotheses consistent with a given set of training examples, delimited by the *specific boundary* and the *general boundary*. In existing studies [4, 5, 3] a hypothesis is a conjunction of attribute-value pairs, which is shown to have limited expressive power [6].

In this paper we investigate version space in a more expressive hypothesis space, where a hypothesis is a *hyperrelation*, which is in effect a disjunction of conjunctions of disjunctions of attribute-value pairs. We propose to use an inductive bias, *E-set*, which turns our attention to equilabelled, supported, and maximal hypertuples. We characterise version space in such a hypothesis space under this bias and show the relationship between the specific boundary and general boundary with respect to *unequivocal data*, a special subset of the data space. We present experimental results on some public datasets.

Keywords

data mining, version space, lattice machine, hypertuple, hyperrelation

1. INTRODUCTION

Version space is a useful and powerful concept in the area of concept learning: A version space is the set of all hypotheses in a hypothesis space consistent with a dataset delimited by the *specific boundary* and the *general boundary* – the sets of most specific and most general hypotheses respectively, to be explained below.

The ELIMINATE-CANDIDATE algorithm [4, 5] is the flagship algorithm used to construct a version space from a dataset. The hypothesis space used in this algorithm is the conjunction of attribute-value pairs, and it is shown to have limited expressive power [6]. It is shown by [2] that the size of the general boundary can grow exponentially in the number of training examples, even when the hypothesis space consists of simple conjunctions of attribute-value pairs.

Table 1 was used in [6] to show the limitation of Mitchell's representation. It was shown that the ELIMINATE-CANDIDATE algorithm can not come up with a consistent specific boundary or general boundary for this example due to this limitation (i.e., the chosen hypothesis space). A possible solution, by increasing the expressive power of the representation, was analysed (extending the hypothesis space to disjunctions of conjunctions of attribute-value pairs from the restrictive conjunctions of attribute-value pairs). But it turned out that, without proper inductive bias, the specific boundary would *always* be overly specific (the disjunction of the observed positive examples) and the general boundary would *always* be overly general (the negated disjunction of the observed negative examples). The desired inductive bias, however, has not been explored.

	Sky	ATemp	Humid	Wind	Water	FCast	d
1	Sunny	Warm	Normal	Strong	Cool	Change	1
2	Cloudy	Warm	Normal	Strong	Cool	Change	1
3	Rainy	Warm	Normal	Strong	Cool	Change	0

Table 1: *Limitation of version space*

In this paper we report a study on version space in a more expressive hypothesis space, where a hypothesis is a set of hypertuples – a generalisation of traditional relational tuples – which is in effect a disjunction of conjunctions of disjunctions of attribute-value pairs. For a given problem domain the set of all hypertuples is a lattice, called *domain lattice*. Lattice machine was introduced in [8, 9] as an approach to supervised learning, which is aimed at finding, as a model of data, a set of hypertuples that are *equilabelled*, *supported*, and *maximal*. Our study shows that, if this restriction is imposed as an inductive bias on the hypotheses in a domain lattice, the version space is a generalisation of Mitchell's version space. This paper reports our findings in this study from a theoretical point of view.

2. DEFINITIONS AND NOTATION

2.1 Order and lattices

A *partial order* on a set U is a binary relation on U which is reflexive, antisymmetric, and transitive.

If U has a largest (smallest) element with respect to a

*A full version of this paper can be found at 193.61.130.50/~cbcj23/papers/vs.html

SAC 2002, Madrid, Spain

partial order $\leq$, we denote it by 1_U (0_U), or just by 1 (0), if U is understood.

Suppose that $\emptyset \neq T \subseteq U$. An object $x \in T \setminus \{1\}$ is called *maximal in* T, if for all $y \in T \setminus \{1\}$, $x \leq y$ implies $x = y$. The set of all maximal elements of T is denoted by $\max T$. Dually, we define *minimal elements* and $\min T$.

We let $\downarrow T \stackrel{\text{def}}{=} \{y \in U : (\exists x \in T)\ y \leq x\}$. If $T = \{a\}$, we will write $\downarrow a$ instead of $\downarrow \{a\}$; more generally, if no confusion can arise, we shall identify singleton sets with the element they contain.

A *semilattice* $\mathcal{L}$ is a nonempty partially ordered set such that for each $x, y \in \mathcal{L}$ the least upper bound $x + y$ exists. For $A \subseteq \mathcal{L}$, we denote the least upper bound of A by $\text{lub}(A)$. $[A]$ is the sub-semilattice of $\mathcal{L}$ generated by A, i.e. $[A]$ is the smallest set which contains A and is closed with respect to $+$.

For $A, B \subseteq \mathcal{L}$, we say that B *covers* A, written as $A \preccurlyeq B$ if for each $s \in A$ there is some $t \in B$ such that $s \leq t$.

For unexplained notation and background reading in lattice theory, we invite the reader to consult [1].

2.2 Information systems

A widely used representation of information is an Object $\mapsto$ Attribute system. An example is the relational data tables.

An *information system* is a tuple $\mathcal{I} = \langle U, \Omega, V_a \rangle_{a \in \Omega}$, where

1. $U = \{x_0, \ldots, x_N\}$ is a nonempty finite set.
2. $\Omega = \{a_0, \ldots, a_T\}$ is a nonempty finite set of mappings $a_i : U \to V_{a_i}$.

We also define $\mathbf{V} \stackrel{\text{def}}{=} \prod_{a \in \Omega} V_a$ and $\mathcal{L} \stackrel{\text{def}}{=} \prod_{a \in \Omega} 2^{V_a}$.

We interpret U as a set of objects and Ω as a set of attributes or features, each of which assigns to an object x a unique value under the respective attribute. If $t \in \mathbf{V}, a \in \Omega, v \in V_a$, then $\langle a, v \rangle$ is called a *descriptor*, and $t(a)$ denotes the projection of t onto a.

For each $x \in U$, we let $\Omega(x) \stackrel{\text{def}}{=} \langle a(x) \rangle_{a \in Q}$; each $\Omega(x)$ is called a *simple tuple*, and the collection of all such tuples is denoted by $\mathbf{D}$. Clearly $\mathbf{D} \subseteq \mathbf{V}$. We can think of $\mathbf{D}$ as a data table with rows being the elements of $\mathbf{D}$, and the columns labelled by the attributes. Any set of simple tuples is called a *simple relation*.

Each element of $\mathcal{L}$ is called a *hypertuple*, and each subset of $\mathcal{L}$ is called a *hyperrelation*. There is a natural embedding of $\mathbf{V}$ into $\mathcal{L}$ by assigning

$$\langle e_0, e_1, \cdots, e_T \rangle \mapsto \langle \{e_0\}, \{e_1\}, \ldots, \{e_T\} \rangle,$$

for each $\langle e_0, e_1, \cdots, e_T \rangle \in \mathbf{V}$. We shall identify $\mathbf{V}$ with the image of this embedding; in particular, we assume that $\mathbf{D} \subseteq \mathbf{V} \subseteq \mathcal{L}$.

$\mathcal{L}$ is a semilattice (in fact a lattice, but we do not need this) under the ordering

$$t \leq s \iff (\forall a \in \Omega)[t(a) \subseteq s(a)]. \tag{1}$$

The least upper bound of t and s is

$$t + s = \langle t(a) \cup s(a) \rangle_{a \in \Omega}. \tag{2}$$

$\mathcal{L}$ is called *domain lattice* for $\mathcal{D}$.

A *decision system* $\mathcal{D}$ is a pair $\langle \mathcal{I}, d \rangle$, where $\mathcal{I}$ is an information system as above, and $d : \mathbf{D} \twoheadrightarrow V_d = \{d_0, \ldots, d_K\}$ is an onto mapping, called a *labelling* of $\mathbf{D}$; the value $d(t)$ is called the *label of* t. We will also refer to d as the *decision attribute*. The mapping d induces a partition $\mathcal{P}_d$ of $\mathbf{D}$ with the classes $\{\mathbf{D}_0, \ldots, \mathbf{D}_K\}$, where for each $i \leq K$, $t \in \mathbf{D}_i \iff d(t) = d_i$. If $\mathcal{P}_d$ has just two elements, we call d a *binary attribute* or a *concept*.

In the sequel, we shall use $\langle \mathcal{I}, d \rangle$ with $\mathbf{V}$ and $\mathbf{D}$ given above as a generic decision system. We call $\mathbf{D}$ the *set of observed examples* or *training data*, and $\mathbf{V}$ is called the *data space*.

2.3 Lattice machine

We recall some facts from [9]. Suppose we have a decision system $\mathcal{D} = \langle \mathcal{I}, d \rangle$, with a training set $\mathbf{D}$ of labelled examples as described in Section 2.2.

We call an element $r \in \mathcal{L}$ *equilabelled* with respect to the class $\mathbf{D}_q$, if $\emptyset \neq \ \downarrow r \cap \mathbf{D} \subseteq \mathbf{D}_q$. In other words, r is equilabelled if $\downarrow r$ intersects $\mathbf{D}$, and there is some $q \leq K$ such that every element in this intersection is labelled d_q; in this case, we say that r *belongs to* $\mathbf{D}_q$.

We denote the set of all equilabelled elements belonging to $\mathbf{D}_q$ by $\mathcal{E}_q$, and let $\mathcal{E} = \bigcup_{q \leq K} \mathcal{E}_q$ be the set of all equilabelled elements. Note that $\mathbf{D} \subseteq \mathcal{E}$, and that $q, r \leq K, q \neq r$ implies $\mathcal{E}_q \cap \mathcal{E}_r = \emptyset$, since $\{\mathbf{D}_0, \ldots, \mathbf{D}_K\}$ partitions $\mathbf{D}$. Recall that for $A \subseteq \mathcal{L}$, $[A]$ is the sub-semilattice of $\mathcal{L}$ generated by the elements of A. We now define

$$\mathbf{E}(A) \stackrel{\text{def}}{=} \{t : t \text{ is maximal in } [A] \cap \mathcal{E}\}.$$

$\mathbf{E}(A)$ is called the *E-set for* A. In other words, for $t \in \mathbf{E}(A)$, we have $t = \text{lub}(X)$ for some $X \subseteq A$, and t is equilabelled. $\mathbf{E}(A)$ contains all equilabelled elements that are maximal, and supported by A.

The special E-set, $\mathbf{E}(\mathbf{D})$, is a model of data $\mathbf{D}$, which is targeted by the lattice machine. This model is, as discussed above, equilabelled, supported, and maximal. An example of such a model is shown in Figure 1.

The labelling function d can be extended over all of $\mathcal{L}$ by setting $d(r) = d_q$ if r is equilabelled and there is $t \in \mathbf{D}$ such that $t \leq r$ and $d(t) = d_q$; $d(r) = \textit{unknown}$ otherwise.

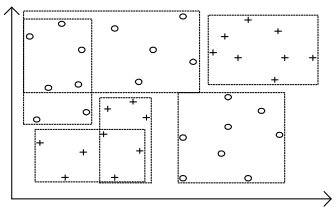

Figure 1: ***A lattice machine model of data. Each rectangle represents a hypertuple, which is equilabelled, supported and maximal.***

For later use, we observe the following fact, the easy proof of which is left to the reader:

Lemma 2.1. *$A \subseteq B \subseteq \mathcal{L}$ implies $\mathbf{E}(A) \preccurlyeq \mathbf{E}(B)$.*

3. VERSION SPACE

The situation in [6] can be described as a special decision system where $d : \mathbf{V} \to \{0, 1\}$, and it is called the *target concept*. The set of positive examples is denoted by D_1, and that of negative examples by D_0. We will describe the

concepts introduced there in our notation and we will follow the explanation in [6, p. 22, Table 2.2].

A *hypothesis* is a hypertuple $t \in \mathcal{L}$ such that for all $a \in \Omega$, $|t(a)| \leq 1$ or $t(a) = V_a$. Thus, for each $a \in \Omega$ we have $t(a) = \emptyset$, or m, or V_a for some $m \in V_a$. The set of all hypotheses is denoted by H. Observe that H is a hyperrelation, and that H is partially ordered by $\leq$ as defined by (1), and that $\mathbf{V} \subseteq H$. If $s, t \in H$, and $s \leq t$, we say that *s is more specific than t* or, equivalently, that *t is more general than s*. We say that *s satisfies the hypothesis t*, if s is a simple tuple (i.e. $s \in \mathbf{V}$) and $s \leq t$. We denote the set of all (simple) tuples that satisfy t by $\text{sat}(t)$. Observe that $\text{sat}(t) = \downarrow t \cap \mathbf{V}$.

More generally, for $A \subseteq \mathcal{L}$ we let $\text{sat}(A) = \downarrow A \cap \mathbf{V}$. If $t(a) = \emptyset$ for some $a \in \Omega$, then t cannot be satisfied. We interpret $s \in \text{sat}(t)$ as "instance s is classified by hypothesis t". According to [6], t is more general than s, if any instance classified by s is also classified by t. That our notion captures this concept is shown by the following result, the easy proof of which is left to he reader.

THEOREM 3.1. *Suppose that $s, t \in H$. Then,*

$$s \leq t \Longleftrightarrow \text{sat}(s) \subseteq \text{sat}(t).$$

Since we are interested in hypotheses whose satisfiable observations are within one class of d, we say that $t \in H$ is called *consistent with* $\langle \mathbf{D}, d \rangle$, if $\downarrow t \cap \mathbf{D} = D_1$. Thus, in this case, the training examples satisfying t are exactly the positive ones. The *version space* $\mathbf{VSp}$ is now the set of all hypotheses consistent with $\langle \mathbf{D}, d \rangle$. In other words,

$$\mathbf{VSp} = \{t \in H : (\forall s)[s \in \text{sat}(t) \cap \mathbf{D} \Longleftrightarrow d(s) = 1]\}.$$

The *general boundary* G is the set of maximal members of H consistent with $\langle \mathbf{D}, d \rangle$, i.e. $G = \max \mathbf{VSp}$. The *specific boundary* S is the set of minimal members of H consistent with $\langle \mathbf{D}, d \rangle$, i.e., $S = \min \mathbf{VSp}$.

The two boundaries delimit the version space in the sense that for any consistent hypothesis t in the version space there are $g \in G$ and $s \in S$ such that $s \leq t \leq g$.

The example in Table 2 illustrates the idea of version space. We follow [6] in writing ? in column a, if $t(a) = V_a$.

Sky	ATemp	Humid	Wind	Water	FCast	d
Training data $\mathbf{D}$						
Sunny	Warm	Normal	Strong	Warm	Same	1
Sunny	Warm	High	Strong	Warm	Same	1
Rainy	Cold	High	Strong	Warm	Change	0
Sunny	Warm	High	Strong	Cool	Change	1
Hypotheses						
Sunny	Warm	?	Strong	?	?	1
Sunny	?	?	Strong	?	?	1
Sunny	Warm	?	?	?	?	1
?	Warm	?	Strong	?	?	1
Sunny	?	?	?	?	?	1
?	Warm	?	?	?	?	1
Specific boundary						
Sunny	Warm	?	Strong	?	?	1
General boundary						
Sunny	?	?	?	?	?	1
?	Warm	?	?	?	?	1

Table 2: *Mitchell's training data, hypotheses and boundaries*

4. THE VERSION SPACE IN A DOMAIN LATTICE

In Mitchell's discussion of version space there are only two decision classes and each hypothesis is a special hypertuple. Now we relax these two restrictions to consider version space in a general domain lattice.

We take a hypothesis to be a hyperrelation, and hence the *hypothesis space* is the set of all hyperrelations in a domain lattice. Thus hypertuples are constructs of hypotheses rather than hypotheses themselves. With this definition of hypothesis, a dataset is a hypothesis for itself and the specific boundary is simply the dataset. Clearly such a hypothesis is uninteresting since it has no power of generalisation. We then need an inductive bias to limit our choice of hypothesis. We adopt the E-set as our inductive bias.

DEFINITION 4.1. *Let P be such that $\mathbf{D} \subseteq P \subseteq \mathbf{V}$. A* hypothesis *$H$ for $\mathbf{D}$ with respect to P is $H \subseteq \mathcal{L}$ such that $H \subseteq \text{E}(P)$. In other words, each $h \in H$ is equilabelled and maximal in $[P] \cap \mathcal{E}$.*

For a hypothesis H to be consistent with $\langle \mathbf{D}, d \rangle$, H should cover $\mathbf{D}$. In other words a *consistent hypothesis* is $H \subseteq \mathcal{L}$ such that $\mathbf{D} \subseteq \text{sat}(H)$ and $H \subseteq \text{E}(P)$. In the sequel we only consider consistent hypothesis.

It is clear that $\text{E}(\mathbf{D})$ and $\text{E}(\mathbf{V})$ are both hypotheses. It follows from Lemma 2.1 that $\text{E}(\mathbf{D}) \preccurlyeq \text{E}(P) \preccurlyeq \text{E}(\mathbf{V})$ for $\mathbf{D} \subseteq P \subseteq \mathbf{V}$. Therefore we call $\text{E}(\mathbf{D})$ *least E-set* and $\text{E}(\mathbf{V})$ *greatest E-set*, denoted by $\mathbb{S}$ and $\mathbb{G}$ respectively. It is not hard to see that $\mathbb{G}$ is the set of all maximal elements of $\mathcal{E}$.

Clearly a hypothesis for $\mathbf{D}$ with respect to P is a subset of $\text{E}(P)$, and the union of all such hypotheses is $\text{E}(P)$. We set $\text{GEN}(\mathbf{D}) = \bigcup_{\mathbf{D} \subseteq P \subseteq \mathbf{V}} \text{E}(P)$.

Clearly $\text{GEN}(\mathbf{D})$ is the set of all hypertuples (constructs) based on which the hypotheses for $\mathbf{D}$ are constructed.

Consider two hypotheses H^j and H^k for $\mathbf{D}$. We say that H^j is *more-general-than* H^k if and only if $H^k \preccurlyeq H^j$. Now Theorem 3.1 implies $\text{sat}(H^k) \subseteq \text{sat}(H^j) \Longleftrightarrow H^k \preccurlyeq H^j$. This implies that our notion captures the concept in Mitchell's original sense.

Mitchell's version space was defined to be the set of all consistent hypotheses. Since, in our work, a hypothesis is a set of hypertuples rather than a single tuple (simple or hyper) and we are interested in hypotheses under the E-set bias, we define version space, specific and general boundaries in terms of hypertuple rather than hypothesis. Formally,

DEFINITION 4.2. Version space in domain lattice *for $\mathbf{D}$ is the set of all hypertuples upon which hypotheses for $\mathbf{D}$ are constructed. Clearly the version space is* $\text{GEN}(\mathbf{D})$.

The general boundary *for $\mathbf{D}$ is $\{h \in \text{GEN}(\mathbf{D})$: there is no $h' \in \text{GEN}(\mathbf{D})$ such that $h' \succ h\}$, and the* specific boundary *for $\mathbf{D}$ is $\{h \in \text{GEN}(\mathbf{D})$: there is no $h' \in \text{GEN}(\mathbf{D})$ such that $h' \prec h\}$.*

The following lemma [1] characterises the specific and general boundaries.

LEMMA 4.3. *The specific boundary for $\mathbf{D}$ is $\mathbb{S}$, and the general boundary for $\mathbf{D}$ is $\mathbb{G}$.*

The following theorem shows that the general boundary covers the whole data space.

THEOREM 4.1. *Let $\mathbb{G} = \{\mathbb{G}_0, \cdots, \mathbb{G}_T\}$. For any $t \in \mathbf{V}$, there must be i such that $t \preccurlyeq \mathbb{G}_i$.*

[1] Due to lack of space we omit all proofs, which can be found in the full paper.

Now we compare our hypothesis space with Mitchell's in terms of their expressive power. Consider an information system $\mathcal{I} = \langle U, \Omega, V_a \rangle_{a\in\Omega}$, based on which the dataset $\mathbf{D}$ is defined. Suppose all attributes $a \in \Omega$ are discrete, and all V_a are finite. In our hypothesis space each attribute $a \in \Omega$ takes on a subset of V_a, so there are $2^{|V_a|}$ different subsets altogether. As a result there are $\prod_{a\in\Omega} 2^{|V_a|}$ different hypertuples. Since a hypothesis is a set of hypertuples (i.e., a hyperrelation), there are $2^{\prod_{a\in\Omega} 2^{|V_a|}}$ distinct hypotheses.

In Mitchell's hypothesis space each attribute $a \in \Omega$ takes on a single value in V_a plus two other special values, "?" and "$\emptyset$". Therefore there are $|V_a| + 2$ different values, and $\prod_{a\in\Omega}(|V_a|+2)$ different tuples. In his conjunctive hypothesis representation, each hypothesis is a single tuple, so there are $\prod_{a\in\Omega}(|V_a| + 2)$ distinct hypotheses. In his disjunctive hypothesis representation, each hypothesis is a set of tuples, so there are $2^{\prod_{a\in\Omega}(|V_a|+2)}$ distinct hypotheses.

Clearly our hypothesis space can represent more distinct objects than Mitchell's can. In this sense we say our hypothesis is more expressive than Mitchell's.

Note that $\mathcal{E}$ characterises eligible hypertuples used to construct hypotheses. So Mitchell's version space can be specialised from our version space in the following way:

- The $\mathcal{E}$ is restricted to $\mathcal{E}_M = \{\gamma(t) : t \in \mathcal{E}, \mathbf{D} \preccurlyeq t\}$, where γ is an operation to transform a hypertuple in such a way that for $t = \langle t_0, \cdots, t_T \rangle$, $\gamma(t) = \langle t'_0, \cdots, t'_T \rangle$, where

$$t'_i = \begin{cases} t(a_i), & \text{if } |t(a_i)| = 1, \\ ?, & \text{otherwise.} \end{cases}$$

  Note that $t(a_i)$ is the projection of tuple t onto a_i.

- There are two classes, i.e., $K = 1$;

- The version space is built for one (positive) class.

Given the above restrictions, the specific boundary is $\mathbb{S} = \{\mathbb{S}_0, \mathbb{S}_1\}$, where $\mathbb{S}_0$ is the specific boundary for the negative class and $\mathbb{S}_1$ is the specific boundary for the positive class. Specifically $\mathbb{S}_i = \text{lub}(\mathbf{D}_i)$, for $i = 0, 1$. Recall that lub is the operation for least upper bound.

Consider the data in Table 3(a). The least and greatest E-sets are shown in Table 3(b) and (c).

X_1	X_2	d
a	0	α
a	1	α
b	0	β

(a)

2^{X_1}	2^{X_2}	d
$\{a\}$	$\{0,1\}$	α
$\{b\}$	$\{0\}$	β

(b)

2^{X_1}	2^{X_2}	d
$\{a\}$	?	α
$\{b\}$	?	β

(c)

Table 3: ***(a) A decision system. (b) The least E-set. (c) The greatest E-set.***

Consider Mitchell's example in Table 2. In our hypothesis space the specific boundary is

Sky	ATemp	Humid	Wind	Water	FCast	d
Sunny	Warm	{Normal, High}	Strong	{Warm, Cool}	{Same, Change}	1
Rainy	Cold	High	Strong	Warm	Change	0

and the general boundary is

Sky	ATemp	Humid	Wind	Water	FCast	d
Sunny	?	?	?	?	?	1
?	Warm	?	?	?	?	1
?	?	Normal	?	?	?	1
?	?	?	?	Cool	?	1
?	?	?	?	?	Same	1
Rainy	?	?	?	?	?	0
?	Cold	?	?	?	?	0
?	?	?	?	Warm	Change	0

For another example in Table 1, the specific boundary is

Sky	ATemp	Humid	Wind	Water	FCast	d
{Sunny,Cloudy}	Warm	Normal	Strong	Cool	Change	1
Rainy	Warm	Normal	Strong	Cool	Change	0

and the general boundary is

Sky	ATemp	Humid	Wind	Water	FCast	d
{Sunny,Cloudy}	?	?	?	?	?	1
Rainy	?	?	?	?	?	0

5. THE RELATIONSHIP BETWEEN THE LEAST AND GREATEST E-SETS

In this section we show the relationship between the two boundaries. Let the least and greatest E-sets be $\mathbb{S}$=$\{$ $\mathbb{S}_0$, $\cdots$, $\mathbb{S}_q$, $\cdots$, $\mathbb{S}_T$ $\}$ and $\mathbb{G}$=$\{\mathbb{G}_0, \cdots, \mathbb{G}_q, \cdots, \mathbb{G}_T\}$, where $\mathbb{S}_q$ is a set of hypertuples and it is the least E-set for class q, and $\mathbb{G}_q$ is the greatest E-set for class q. We note that $((\downarrow \mathbb{S}_i) \cap \mathbf{D}) \cap ((\downarrow \mathbb{S}_j) \cap \mathbf{D}) = \emptyset$, but $(\downarrow \mathbb{S}_i) \cap (\downarrow \mathbb{S}_j) \neq \emptyset$. This is the same for $\mathbb{G}$. Assume that our classification rule is: for any $t \in \mathbf{V}$, if $t \preccurlyeq \mathbb{S}_i$ (or $t \preccurlyeq \mathbb{G}_i$) then t is labelled as $d_{\mathbb{S}}(t)$ (or $d_{\mathbb{G}}(t)$). Under this rule it is possible that a tuple may have two or more different labels since it may be covered by two or more hypertuples with different labels. This situation happens with Mitchell's version space as well [6], where the problem was solved by majority voting. In this section we examine only those simple tuples which have unique labels in the greatest E-set (hence unique labels in the least E-set as well).

Formally, let $\mathbf{V}_{\mathbb{G}_q} = (\downarrow \mathbb{G}_q \setminus \bigcup_{i\neq q} \downarrow \mathbb{G}_i) \cap \mathbf{V}$. Let $\mathbf{V}' = \bigcup_i \mathbf{V}_{\mathbb{G}_i}$. Each $t \in \mathbf{V}'$ is a simple tuple and it is covered by one and only one $\mathbb{G}_i$. We call $\mathbf{V}'$ the *unequivocal dataset* with respect to the greatest E-set.

THEOREM 5.1. *Consider $\mathbb{S}_q$ and $\mathbb{G}_q$, the least and greatest E-sets for class q. Let $\mathbb{S}'_q = \{x \in \mathcal{L} : x \not\preccurlyeq \mathbb{S}_i, i \neq q\}$. Let $\mathbf{V}_{\mathbb{S}'_q} = \{t \in \mathbf{V}' : \exists x \in \mathbb{S}'_q \cap \mathcal{E}$ such that $t \leq x\}$. Then $x \in \mathbf{V}_{\mathbb{G}_q}$ if and only if $x \in \mathbf{V}_{\mathbb{S}'_q}$.*

This theorem implies that, for unequivocal data $t \in \mathbf{V}'$, to see whether $t \preccurlyeq \mathbb{G}_i$ we only need to check whether $t \preccurlyeq \mathcal{E}$ and $t \not\preccurlyeq \mathbb{S}_j$ for all $j \neq i$. This theorem establishes a relationship between the specific and general boundaries with respect to unequivocal data. This theorem also explains the C2 algorithm for classification used in [9].

Consider an example in Figure 2. Suppose the least E-set is $\mathbb{S} \stackrel{\text{def}}{=} \mathbb{S}_0 \cup \mathbb{S}_1 = \{h_0, h_1\}$. We can, for example, classify t_2 according to the above theorem in the following way. To see if $t_2 \preccurlyeq \mathbb{G}_1$, we need to check whether t_2 is NOT covered by h_0 (i.e., t is outside the left rectangle) and whether t_2 is covered by another rectangle that overlaps h_1 but not h_0 (hence this rectangle is equilabelled). This is clearly true so $t_2 \preccurlyeq \mathbb{G}_1$ and we can therefore classify t_2 by h_1. However $t_2 \not\preccurlyeq \mathbb{G}_0$ so we can not classify t_2 by h_0.

THEOREM 5.2. *Consider $\mathbb{S}_q$ and $\mathbb{G}_q$. For $t \in \mathbf{V}'$, $t \preccurlyeq \mathbb{G}_q$ if and only if there is $h \in \mathbb{S}_q$ such that $t + h$ is equilabelled.*

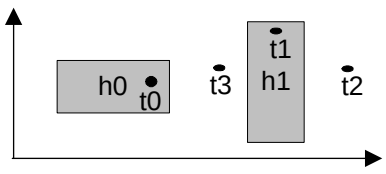

Figure 2: *h_0 and h_1 are equilabelled, supported and maximal hypertuples in different classes. t_0, t_1, t_2 and t_3 are simple tuples to be classified, where t_0, t_1 and t_2 are unequivocal and t_3 is not.*

For an unequivocal datum $t \in \mathbf{V}'$ there must be q such that $t \in \mathbf{V}_{\mathbb{G}_q}$ according Theorem 4.1, and t can then be classified by $\mathbb{G}_q$. Theorem 5.2 suggests that, to find such q, we only need to examine the least E-set to find an $h \in \mathbb{S}_i$ such that $t + h$ is equilabelled. Such an h is guaranteed to exist. This theorem establishes another important relationship between the specific and general boundaries,

6. EVALUATION

Algorithms can be designed to exploit Theorems 5.1 and 5.2 for classification. We wrote a C++ program called *generalised lattice machine* or *GLM* for short, which implements Theorem 5.1. For efficiency reason we distinguish between two types of unequivocal data: those that are covered by a hyper tuple and those that are not. We call the former *primary data* and the latter *secondary data*.

We carried out some experiments using public data. The public datasets are described in Table 4, which are available from UC Irvine Machine Learning Repository. The experimental results are also shown in Tables 4, along with C5.0 results on the same datasets. It is clear from the result that primary (hence unequivocal) data account for over 70% of all data and the algorithm performed significantly better on primary data than C5.0 on all data.

Dataset	#Attr.	#Exa.	Pred. success (%)				%PP
			C5.0	SR	PSR	SSR	
Annealing	38	798	96.6	96.4	98.0	62.9	92.5
Australian	14	690	90.6	95.1	95.1	100.0	87.2
Auto	25	205	70.7	82.4	87.0	63.0	56.1
Diabetes	8	768	72.7	70.7	71.0	40.0	66.8
German	20	1000	71.7	72.6	72.6	N/A	65.4
Glass	9	214	80.4	86.6	87.6	76.5	79.4
Heart	13	270	77.0	81.9	81.9	N/A	61.5
Hepatitis	19	155	80.6	82.9	84.1	50.0	69.0
Horse-Colic	22	368	85.1	82.4	82.7	N/A	67.7
Iris	4	150	94.7	93.1	97.6	66.7	82.7
Sonar	60	208	71.6	81.6	81.3	100.0	59.1
TTT	9	958	86.2	96.2	96.1	100.0	94.9
Vote	18	232	96.5	96.5	98.6	77.3	89.7
Wine	13	178	94.3	99.0	99.0	N/A	53.9
Average			83.5	87.0	88.0	73.7	76.1

Table 4: ***General information about the datasets and the prediction success of C5.0 on all data and of GLM on unequivocal data. The validation method used is 5 fold cross validation. The acronyms are: SR – overall success ratio, PSR – success ratio of primary data, SSR – success ratio of secondary data, PP –the percentage of primary data, and N/A – not available.***

7. DISCUSSION AND CONCLUSION

Mitchell's classical work on version space has been followed by many. Most notably Hirsh and Sebag. Hirsh [3] discusses how to merge version spaces when a central idea in Mitchell's work is removed – a version space is the set of concepts *strictly consistent* with training data. This merging process can therefore accommodate noise. Sebag [7] presents what she calls a disjunctive version space approach to learning disjunctive concepts from noisy data. A separate version space is learned for each positive training example, then new instances are classified by combining the votes of these different version spaces.

In this paper we investigated version spaces in a more expressive hypothesis space – the domain lattice. Without a proper inductive bias the version space is uninteresting. We show that, with E-set (equilabelled, supported and maximal) as an inductive bias this version space is a generalisation of Mitchell's original version space, which employs a different type of inductive bias. We also show that the specific boundary and general boundary are equivalent with respect to unequivocal data under a special operation.

The boundaries can be used as models of data, which can then be used to classify unequivocal data. Experimental results show that this classification approach on unequivocal data has significantly higher classification accuracy than C5.0 on all data.

8. REFERENCES

[1] G. Grätzer. *General Lattice Theory*. Birkhäuser, Basel, 1978.

[2] D. Haussler. Quantifying inductive bias: Ai learning algorithms and valiant's learning framework. *Artificial Intelligence*, 36:177–221, 1988.

[3] H. Hirsh. Generalizing version spaces. *Machine Learning*, 17(1):5–46, 1994.

[4] T. M. Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proc. 5th IJCAI*, pages 305–310, 1977.

[5] T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18, 1982.

[6] T. M. Mitchell. *Machine Learning*. The McGraw-Hill Companies, Inc, 1997.

[7] M. Sebag. Delaying the choice of bias: A disjunctive version space approach. In *Proc. of the 13th International Conference on Machine Learning*, pages 444–452, San Francisco, 1996. Morgan Kaufmann.

[8] H. Wang, W. Dubitzky, I. Düntsch, and D. Bell. A lattice machine approach to automated casebase design: Marrying lazy and eager learning. In *Proc. IJCAI99*, pages 254–259, Stockholm, Sweden, 1999.

[9] Hui Wang, Ivo Düntsch, and Günther Gediga. Classificatory filtering in decision systems. *International Journal of Approximate Reasoning*, 23:111–136, 2000.