Classificatory Filtering in Decision Systems

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†
Fachbereich Psychologie/Methodenlehre
Universität Osnabrück
49069 Osnabrück, Germany
ggediga@Luce.Psycho.Uni-Osnabrueck.DE

Abstract

Classificatory data filtering is concerned with reducing data in size while preserving classification information. Düntsch and Gediga [2] presented a first approach to this problem. Their technique collects values of a single feature into a single value. In this paper we present a novel approach to classificatory filtering, which can be regarded as a generalisation of Düntsch and Gediga's approach. This approach is aimed at collecting values of a set of features into a single value. We look at the problem abstractly in the context of lattice. We focus on *hypergranules* (arrays of sets) in a problem domain, and it turns out the collection of all hypergranules is a lattice. Our solution (namely LM algorithm) is formulated to find a set of maximal elements for each class, which covers all elements in a given dataset and is consistent with the dataset. This is done through the lattice sum operation. In terms of decision systems, LM collects attributes values while preserving classification structure.

To use the filtered data for classification, we present and justify two measures (C^0 and C^1) for the relationship between two hypergranules. Based on the measures, we propose an algorithm (C2) for classification.

Both algorithms are evaluated using real world datasets and are compared with C4.5. The result is analysed using statistical test methods and it turns out that there is no statistical difference between the two. Regression analysis shows that the reduction ratio is a strong indicator of prediction success.

Keywords: artificial intelligence, machine learning, rough set, data filtering, data reduction, decision system, lattice.

1 Introduction

Data reduction is a process which is used to transform raw data into a more condensed form without losing significant semantic information. In data mining, data reduction in a stricter sense refers to

*Corresponding author
†Equal authorship implied

feature selection and data sampling [15]; in a broader sense, data reduction can be regarded as the main task of data mining [4].

Horizontal reduction consists of identifying several rows in a data table according to specified criteria. The identification of suitable rows has the welcome effect of strengthening rules in the following sense: If the prediction of a decision attribute is based on a few values only, the statistical significance of the rule may be low, and it cannot be ruled out that the rule is due to chance; data reduction may enhance the statistical basis of the rule, and thus increase its significance [1].

Discretization of continuous attributes which constructs intervals within data domains and collects attribute values within each of the intervals is a well known device of data analysis and prediction. However, in most discretization methods, parameters outside the given data have to be assumed in order for the procedure to work. The choice of these parameters is largely subjective, and may result in unwelcome decontextualisation. On the other hand, classificatory data filtering as explained below uses only the structural information given by the data under consideration, and does not take into account numerical information of the data domains; neither does it introduce additional parameters. Indeed, it stays on the level of operationalization in the sense of [6], and therefore it can be used as a safe pre-processing mechanism before "harder" computational methods are employed.

A first approach to classificatory data filtering was taken by [2]. This technique collects values of a feature into a single value by taking a union of deterministic equivalence classes which are totally contained in a class of the decision attribute. For example, if we have an attribute q and a rule

$$\text{If } q = 2 \text{ or } q = 3 \text{ or } q = 5 \text{ then } d = \text{blue},$$

then we can collect 2,3,5 into a single attribute value of q.

The important feature of this procedure is that the internal dependency structure of the system is kept intact, and that one does not need additional parameters as other more sophisticated methods.

As an example, consider the famous Iris data. The data used by [5] to demonstrate his discriminant analysis consists of 50 specimen of each of the iris species *Setosa, Versicolor,* and *Virginica*, measured by the features given in Tab. 1.

Table 1: Iris Data

Attribute	Range in mm	Classes	Attribute	Range in mm	Classes
Sepal length	$43 \leq x \leq 79$	35	Petal length	$10 \leq x \leq 69$	23
Sepal width	$22 \leq x \leq 44$	43	Petal width	$1 \leq x \leq 25$	22

The column "classes" tells us, how many of the attribute values are actually taken by the specimen. On inspection of the data, we find, for example, the rules

$$\text{If "Petal width"} \in \{1, \ldots, 6\}, \text{ then "Species"} = \text{Setosa}$$

$$\text{If "Petal width"} \in \{10, \ldots, 13\}, \text{ then "Species"} = \text{Versicolor}$$

$$\text{If "Petal width"} \in \{17, 20, \ldots, 25\}, \text{ then "Species"} = \text{Virginica}$$

If we collect the appropriate values into one single set, then the number of "Petal width" classes is reduced to eight. The complete analysis is given in Tab. 2. There, the new number of values of the attribute is given in brackets, e.g. after collecting non–splitting values into one, Sepal Length takes only 22 values, compared to 35 before. The column # tells us, how many objects are described by this new value; for example, the set $\{10, \ldots, 19\}$ of values of Petal Length determines all of the Setosa class.

Table 2: One dimensional classificatory filtering

Sepal length (22)			Sepal width (16)		
	Filter	#		Filter	#
Setosa	43–48, 53	17	Setosa	35, 37, 39–44	15
Versicolor	66,70	3	Versicolor	20,24	4
Virginica	71–79	12	Virginica	–	–
Petal length (8)			**Petal width (8)**		
	Filter	#		Filter	#
Setosa	10–19	50	Setosa	1–6	50
Versicolor	30–44,46,47	37	Versicolor	10–13	28
Virginica	52, 54–69	34	Virginica	19–25	34

In this paper, we generalize this one dimensional approach to more attributes by allowing sets of attribute values in more than one column as entries in a data table. These *hypergranules* [1] can be made into a semilattice in a natural way, and a hypergranule can represent one or more rows of our data table, according to the relation of their values with respect to a decision attribute.

The paper is structured as follows: In Section 2 we recall some definitions from lattice theory and introduce our notation of data relations and decision systems. Section 3 will provide the formal reduction machinery. Classification based on the filtered data is discussed in Section 4. An example is presented in Section 5 to illustrate both the filtering method and the classification method. The proposed methods are evaluated and the results and analysis are reported in Section 6. In Section 7 related work is discussed and compared. Finally Section 8 summarises and concludes the paper.

[1] The concept of hypergranule or *hyper relation* is first proposed in [12].

2 Definitions and notation

2.1 Order and lattices

A *partial order* on a set P is a binary relation $\leq$ with the properties

$$\begin{aligned} x &\leq x, && \text{(Reflexive)} \\ x \leq y \text{ and } y \leq x &\text{ imply } x = y, && \text{(Antisymmetric)} \\ x \leq y \text{ and } y \leq z &\text{ imply } x \leq z. && \text{(Transitive)} \end{aligned}$$

Suppose that $\mathcal{P} = \langle P, \leq \rangle$ is a partially ordered set and $T \subseteq P$. T is called an *antichain* if any two elements of T are incomparable in $\leq$. We let $\downarrow T \stackrel{\text{def}}{=} \{y \in P : (\exists x \in T)\, y \leq x\}$. If $T = \{a\}$, we will write $\downarrow a$ instead of $\downarrow \{a\}$; more generally, if no confusion can arise, we shall usually identify singletons with the element they contain.

A sup– *semilattice* L is a nonempty partially ordered set such that for each $x, y \in L$ the least upper bound $x + y$ exists. The greatest element of L, if it exists, is denoted by 1; if L is finite then 1 exists, and it is equal to $\sum_{a \in L} a$. An element $a \in L$ is called *maximal*, if $a \neq 1$ and for all $b \in L$,

$$a \lneq b \Rightarrow b = 1.$$

If $A, B \subseteq L$, we write $A \preccurlyeq B$ if for each $s \in A$ there is some $t \in B$ such that $a \leq b$; furthermore, we set $A + B = \{a + b : a \in A, b \in B\}$.

Lemma 2.1. *If $A \preccurlyeq B$, $B \preccurlyeq A$, and both A and B are antichains, then $A = B$.*

Proof. Assume w.l.o.g. that $a \in A \setminus B$. Since $A \preccurlyeq B$, there is some $b \in B$ such that $a \leq b$, and $B \preccurlyeq A$ implies the existence of some $c \in A$ with $b \leq c$. Since $a \notin B$, we have $a \lneq c$, contradicting that A is an antichain. □

For unexplained notation and background reading in lattice theory, we invite the reader to consult [7].

2.2 Decision systems

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

1. $U = \{a_1, \ldots, a_N\}$ is a nonempty finite set.
2. $\Omega = \{x_1, \ldots, x_T\}$ is a nonempty finite set of mappings $x_i : U \to V_{x_i}$.

We interpret U as a set of objects and Ω as a set of attributes or features each of which assigns to an object a its value under the respective attribute. Let $V \stackrel{\text{def}}{=} \Pi_{x\in\Omega} V_x$. For $a \in U$, we let

$$\Omega(a) = \langle x(a)\rangle_{x\in\Omega}, \tag{2.1}$$

Each $\Omega(a)$ is called a *granule*, and the collection of all granules is denoted by D. Clearly $D \subset V$. Thus, if $t \in D$, there is some $a \in U$ such that $\Omega(a) = t$; if $x \in \Omega$, then $t(x)$ is just $x(a)$.

A *decision system* $\mathcal{D}$ is a pair $\langle \mathcal{I}, d\rangle$, where $\mathcal{I} = \langle U, \Omega, V_x\rangle_{x\in\Omega}$ is an information system as above, and $d : D \twoheadrightarrow V_d = \{m_1, \ldots, m_K\}$ is an onto mapping, called a *labeling* of 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 D with the classes $\{M_0, \ldots, M_K\}$, where

$$t \in M_i \Longleftrightarrow d(t) = m_i. \tag{2.2}$$

3 Collecting attribute values

In the sequel, we shall use $\mathcal{D}$ as described above as a generic decision system.

Let $\mathcal{T}$ be the set $\prod_{x\in\Omega} 2^{V_x}$; $\mathcal{T}$ is a $+$ – semilattice (in fact, a Boolean algebra, but we will not need this here) under the ordering

$$t \leq s \Longleftrightarrow t(x) \subseteq s(x)$$

for all $x \in \Omega$. The elements t of $\mathcal{T}$ with $|t(x)| = 1$ for all $x \in \Omega$ are called *simple tuples*. There is a natural embedding of D into $\mathcal{T}$ by assigning

$$\Omega(a) \mapsto \langle \{x_1(a)\}, \{x_2(a)\}, \ldots, \{x_T(a)\}\rangle. \tag{3.1}$$

and we shall identify D with result of this embedding.

It is our aim to reduce the data with respect to the classes of the decision attribute; this can be done one class at a time. Thus, fix a class M in $\mathcal{P}_d$ belonging to $m \in V_d$, and let L_M be the subsemilattice of $\mathcal{T}$ generated by M; the elements of L_M are called *hypergranules*. We call an element $r \in L_M$ *equilabeled* (with respect to M), if $\downarrow r \cap D \subseteq M$. In other words, everything below r which is in D is labeled m. Each equilabeled element may replace a number of elements of M, and thus, we result in some form of data compression.

Let $\mathcal{E}$ be the set of all $r \in L_M$ which are equilabeled w.r.t. M. A *cover of* M is a set $C \subseteq \mathcal{E}$ such that for each $t \in M$ there is some $c \in C$ such that $t \leq c$, i.e. $M \preccurlyeq C$.

Clearly, M is a cover of itself. A less trivial example is the following: If for all $t \in D$ and for some $x \in \Omega$,

$$\text{If } t(x) = a \text{ or } t(x) = b, \text{ then } t \in M,$$

then

$$C = \left\{\sum\{t \in D : t(x) \in \{a, b\}\}\right\} \cup \{t \in D : t(x) \notin \{a, b\}\}.$$

is a cover.

If C is a cover, $s, t \in C$ and $s + t \in \mathcal{E}$, then $(C \setminus \{s, t\}) \cup \{s + t\}$ is also a cover with smaller cardinality, i.e. with greater data reduction. This leads to the following definition: An *E–set* is a cover C for which $s, t \in C$ implies $s + t \notin \mathcal{E}$. E–sets are those covers in which the sum of two elements is not equilabeled with respect to M; in particular, each E–set is an antichain.

A prime candidate for a set of hypergranules which can replace M is the set H of maximal elements of $\mathcal{E}$. Since $M \subseteq \mathcal{E}$ and each element of $\mathcal{E}$ is below or equal to some element of H, we see that H covers M. It is clear that this H is an E-set for M. Therefore our objective becomes, given M, finding an E-set for M.

An algorithm to find H is as follows (LM - algorithm):

1. $C_1 \stackrel{\text{def}}{=} M$.
2. $C_{k+1} \stackrel{\text{def}}{=}$ The set of maximal elements of $[\downarrow (C_k + M)] \cap \mathcal{E}$.

Each C_k is a subset of $\mathcal{E}$, and $C_k \preccurlyeq [\downarrow (C_k + M)] \cap \mathcal{E} \preccurlyeq C_{k+1}$. The finiteness of L_M and the fact that each C_k is an antichain now imply that there is some n such that $C_n = C_{n+1}$, and therefore $C_n = C_r$ for all $r \geq n$.

Claim: $C_n = H$.

Proof. We first show that $(\overbrace{M + \ldots + M}^{i \text{ times}}) \cap \mathcal{E} \preccurlyeq C_i$: This is clearly true for $i = 1$; thus, suppose that it holds for all $1 \leq j < i$. Let $t = t_1 + \ldots + t_i \in \mathcal{E}$. Then, $t_2 + \ldots + t_i \in (\overbrace{M + \ldots + M}^{(i-1)\text{- times}}) \cap \mathcal{E}$, and thus, $t_2 + \ldots + t_i \in C_i$. It follows that $t \in (C_i + M) \cap \mathcal{E}$, and hence, t is below some maximal element of $[\downarrow (C_i + M)] \cap \mathcal{E} = C_i$.

Since $C_n \subseteq \mathcal{E}$, and H is the set of maximal elements of $\mathcal{E}$, we have $C_n \preccurlyeq H$. By Lemma 2.1 it suffices to show that $H \preccurlyeq C_n$; indeed, since $C_k \preccurlyeq C_n$ for all $k \in \omega$ it is enough to show $H \preccurlyeq C_k$ for some k. Thus, let $t \in H$; then, $t \in \mathcal{E}$ and there are $t_0, \ldots, t_k \in M$ such that $t = \bigvee_{i \leq k} t_i$. It follows from the previous result that there is some $s \in C_k$ such that $t \leq s$, which proves our claim. □

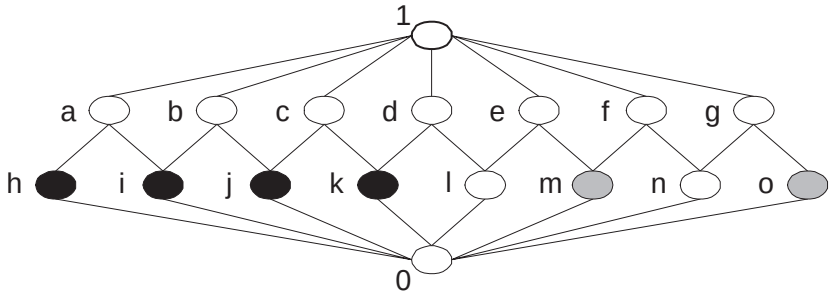


Figure 1: A labelled lattice showing that H is not the least cover of M.

We observe that H need not be the least cover of M in terms of cardinality, since there may be covers $C \subsetneq H$. Consider the lattice in Figure 1. For the dark black class, LM find $H = \{a, b, c\}$. But $\{a, c\}$ is also a cover and has less number of elements.

4 Assigning new information

Suppose we have chosen for each class M_i of $\mathcal{P}$ an E–set E_i, i.e.

1. Each $t \in E_i$ is a sum of elements of D,
2. Each $t \in E_i$ is equilabeled m_i,
3. Each element of M_i is below some $t \in E_i$,
4. If $s, t \in E_i$, then $s + t \notin E_i$.
5. $E_i \cap E_j = \emptyset$ for $i \neq j$, since every element of E_i is equilabeled with m_i.

To label $t \in V$, we have the following three cases:

- Single coverage: $t \in\downarrow E_i$ for one and only one i.
- Multiple coverage: $t \in\downarrow E_i$ for more than one i. In other words, $\downarrow E_i \cap \downarrow E_j = \emptyset$ is not true. Suppose that we have the system given in Table 3. The hypergranules are $\langle\{0, 1\}, \{0, 1\}\rangle$ and $\langle\{0, 2\}, \{0, 2\}\rangle$ for the 0-class and 1-class respectively. Clearly $\langle 0, 0\rangle$ is below both of them.
- Non coverage: $t \notin E_i$ for any i. Due to the incompleteness of the data (decision system), the E-sets may also be incomplete in the sense that they don't cover the whole data space. Therefore it is possible that some $t \in V$ is not covered by any E_i. Consider Table 3 again. Clearly $\langle 2, 1\rangle$ is not below any of the hypergranules.

Our solution to the assignment problem is designed to address each of the above cases:

Table 3: An example

U	p	q	D
a	1	0	0
b	0	1	0
c	2	0	1
d	0	2	1

Single coverage

For $t \in V$, if there is only one E_i such that $t \in\downarrow E_i$, it is reasonable to label t by m_i.

Multiple coverage

For $t \in V$, if there are i and j such that $t \in\downarrow E_i$ and $t \in\downarrow E_j$, then the labelling is determined by whichever has the largest coverage of the elements in D. For example, if $s_0, s_1 \in E_i$, $s_2 \in E_j$, and $t \leq s_0$, $t \leq s_1$ and $t \leq s_2$, then we would label t by m_i instead of m_j. In our experience, however, this case rarely happens in practice.

Non coverage

For $t \in V$, if there is no E_i such that $t \in\downarrow E_i$, we would tend to examine the likelihood of each $s \in \bigcup E_i$ potentially covering this t. By "potentially" we mean that if sufficient information were given in the dataset, t would be covered by s. The t is then labelled by the label of the s with the greatest likelihood. The question now is: how to measure the likelihood of a tuple potentially covered by an E-set? To introduce our solution, we look at the following example first.

Example 1. Let $\Omega = \{X_1, X_2\}$, $V_{X_1} = \{a, b\}$, $V_{X_2} = \{0, 1\}$. The data space $V = V_{X_1} \times V_{X_2}$ is shown in Table 4. The $\mathcal{T}$ is shown in Table 7. Now let Y be a decision attribute where $V_Y = \{\alpha, \beta\}$, and assume that we have a decision system (dataset) as shown in Table 5. Using the algorithm described above, we get two E-sets as shown in Table 6 with one for each class. Clearly tuple $t = u_3 = \langle b, 1\rangle$ in Table 4 is not covered by either E-set. Then uncertainty arises as to how to label t. Looking at the problem tuple-wise, we find that $t(X_1) \not\leq u'_0(X_1)$ but $t(X_2) \leq u'_0(X_2)$; and that $t(X_1) \leq u'_1(X_1)$ but $t(X_2) \not\leq u'_1(X_2)$. This seems that t should be equally likely labelled as either as α or β. Looking at the X_2 column, however, we find that there is the likelihood that 0 and 1 belong to the same class (cluster) of the domain of X_2, whereas the current model (in Table 6) doesn't support putting a and b in the same cluster. Therefore it is more likely that t is labelled as β than as α.

In the spirit of Example 1, we now formally describe our measures for the likelihood of one hypergranule happening given another hypergranule. Given two hypergranules, t_0 and t_1, we first of all need a measure for the likelihood that t_0 is covered by t_1. Since t_0 may not be fully covered by t_1 hence assuming t_0 is covered by t_1 may not preserve the structure in the dataset, we then need a measure for the degree in which assuming this could preserve the structure. Displaying the E-sets in a table (see Table 6), it turns out that each column represents a subset of the power set of the attribute domain. This can be studied in the context of Evidence Theory [8].

Let $X \subseteq \Omega$, and $S \stackrel{\text{def}}{=} V_X$ be the domain of X. Consider a mass function [2] $m : 2^S \rightarrow [0, 1]$ such that $\sum_{x \in 2^S} m(x) = 1$. Given $a, b \in 2^S$, where $m(b) \neq 0$, the first measure is derived by answering this question: what is the likelihood that b appears whenever a appears? In other words, if a appears, what is the likelihood that b will be regarded as appearing as well? Denoting this likelihood by $C^0_X(b|a)$, one solution is:

$$C^0_X(b|a) = \frac{\sum_{a \cup b \subseteq c} m(c)}{\sum_{b \subseteq c} m(c)}$$

$C^0_X(t_1|t_0)$ is the likelihood of $t_0(X) \cup t_1(X)$ appearing relative to the likelihood of $t_1(X)$ appearing.

In the same spirit, another measure is defined as

$$C^1_X(b|a) = \frac{\sum_{c \subseteq b} m(c)}{\sum_{c \subseteq a \cup b} m(c)}$$

$C^1_X(b|a)$ measures the degree in which merging a and b preserves the existing structure embodied by the mass function.

The definition of C^0_X is easy to understand, and the significance of C^1_X can be illustrated as follows.

Example 2. Consider two intervals of the same length in Figure 2. We are interested in two cases: non-overlapping and overlapping, as shown in the figure. Given each of the 5 *points on the top and bottom figures, we now calculate the* C^0 *and* C^1 *values assuming that the mass function is a linear function of interval length:*

$C^0(I_1|t_1) = 0$, $C^1(I_1|t_1) = 1$, $C^0(I_2|t_1) = 0$, $C^1(I_2|t_1) = 0.5$;
$C^0(I_1|t_2) = 1$, $C^1(I_1|t_2) = 1$, $C^0(I_2|t_2) = 0$, $C^1(I_2|t_2) = 1$;
$C^0(I_1|t_3) = 0$, $C^1(I_1|t_3) = 1$, $C^0(I_2|t_3) = 0$, $C^1(I_2|t_3) = 1$;
$C^0(I_1|t_4) = 0$, $C^1(I_1|t_4) = 1$, $C^0(I_2|t_4) = 1$, $C^1(I_2|t_4) = 1$;
$C^0(I_1|t_5) = 0$, $C^1(I_1|t_5) = 0.5$, $C^0(I_2|t_5) = 0$, $C^1(I_2|t_5) = 1$.

[2] In the context of decision table, the mass function can be regarded as the uniform distribution over the tuples in D collapsed to the set of hypergranules. For an example, consider the decision system in Table 5. We can reasonably assume a uniform distribution for the table. Collapsing the tuples in Table 5 as hypergranules in Table 6, we get a new distribution over the hypergranules – $u'_0 : 2/3, u'_1 : 1/3$. Then the mass function for 2^{X_2} becomes $\{0, 1\} : 2/3$, and $\{0\} : 1/3$.

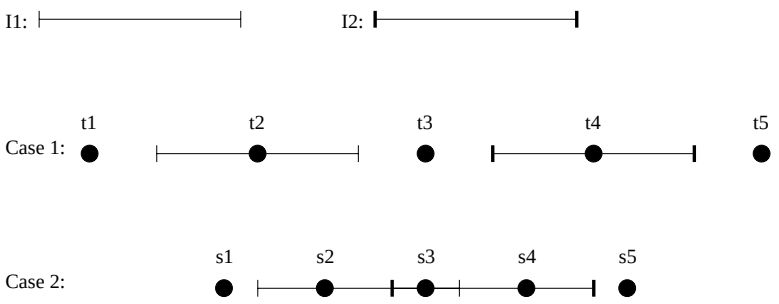


Figure 2: A one dimensional example justifying the significance of the two measures.

$C^0(I_1|s_1) = 0$, $C^1(I_1|s_1) = 1$, $C^0(I_2|s_1) = 0$, $C^1(I_2|s_1) = 0.5$;
$C^0(I_1|s_2) = 1$, $C^1(I_1|s_2) = 1$, $C^0(I_2|s_2) = 0$, $C^1(I_2|s_2) = 1$;
$C^0(I_1|s_3) = 1$, $C^1(I_1|s_3) = 1$, $C^0(I_2|s_3) = 1$, $C^1(I_2|s_3) = 1$;
$C^0(I_1|s_4) = 0$, $C^1(I_1|s_4) = 1$, $C^0(I_2|s_4) = 1$, $C^1(I_2|s_4) = 1$;
$C^0(I_1|s_5) = 0$, $C^1(I_1|s_5) = 0.5$, $C^0(I_2|s_5) = 0$, $C^1(I_2|s_5) = 1$.

Now assume that given any point, one of the interval must be "activated" (i.e., the interval is regarded as appearing accordingly). Then intuitively, for t_1, I_1 other than I_2 should be "activated" since t_1 is closer to I_1 than to I_2. This is reflected by $C^0(I_1|t_1) = 0$, $C^1(I_1|t_1) = 1$, $C^0(I_2|t_1) = 0$, $C^1(I_2|t_1) = 0.5$. For t_2, clearly I_1 should be activated, which is reflected by $C^0(I_1|t_2) = 1$ and $C^0(I_2|t_2) = 0$. For t_3, both I_1 and I_2 are equally likely to be activated, which is reflected by $C^0(I_1|t_3) = 0$, $C^1(I_1|t_3) = 1$, $C^0(I_2|t_3) = 0$, $C^1(I_2|t_3) = 1$. Similar analysis can be done for t_4 and t_5. For s_3, both I_1 and I_2 should be equally likely to be activated, which is reflected by the fact that $C^0(I_1|s_3) = 1$, $C^1(I_1|s_3) = 1$, $C^0(I_2|s_3) = 1$, $C^1(I_2|s_3) = 1$.

From this example, we can draw a two-stage decision rule: consider two intervals I_1 and I_2. Given a point t, if $C^0(I_{i_1}|t) > C^0(I_{i_2}|t)$, then I_{i_1} is more likely to be activated than I_{i_2}, where $\{i_1, i_2\} = \{1, 2\}$; if $C^0(I_{i_1}|t) = C^0(I_{i_2}|t)$ and $C^1(I_{i_1}|t) > C^1(I_{i_2}|t)$, then I_{i_1} is more likely to be activated than I_{i_2}; otherwise, I_1 and I_2 are equally likely to be activated with regard to these two measures. In this case we probably need to resort to other measures to decide which is more likely to be activated.

The above decision rule can be generalised to our case, if the set inclusion relation ($\subseteq$) is replaced by the tuple ordering relation ($\leq$) on page 3.

Example 3. Now let's look at Example 1 again. Assume a uniform distribution for the decision system in Table 5. Given the model in Table 6, we want to classify a new tuple $t =< b, 1 >$. Using the above definitions and letting $X = \{2^{X_1}, 2^{X_2}\}$, we have $C^0_X(u'_0|t) = 0$, $C^0_X(u'_1|t) = 0$; $C^1_X(u'_0|t) = 2/3$, $C^1_X(u'_1|t) = 1$. These measures mean that t is not covered by u'_0 nor u'_1, and that merging t with u'_1 better preserves the structure than merging t with u'_0. Therefore we would classify t as β.

U	X_1	X_2
u_0	a	0
u_1	a	1
u_2	b	0
u_3	b	1

Table 4: *A data space.*

U	X_1	X_2	Y
u_0	a	0	α
u_1	a	1	α
u_2	b	0	β

Table 5: *A decision system.*

U	2^{X_1}	2^{X_2}	Y
u'_0	$\{a\}$	$\{0,1\}$	α
u'_1	$\{b\}$	$\{0\}$	β

Table 6: *The model.*

$2^{V_{X_2}}$	$\emptyset$	$\{0\}$	$\{1\}$	$\{0,1\}$	$\emptyset$	$\{0\}$	$\{1\}$	$\{0,1\}$	$\emptyset$	$\{0\}$	$\{1\}$	$\{0,1\}$	$\emptyset$	$\{0\}$	$\{1\}$	$\{0,1\}$
$2^{V_{X_1}}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$	$\{b\}$	$\{b\}$	$\{b\}$	$\{b\}$	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$
U	u''_0	u''_1	u''_2	u''_3	u''_4	u''_5	u''_6	u''_7	u''_8	u''_9	u''_{10}	u''_{11}	u''_{12}	u''_{13}	u''_{14}	u''_{15}

Table 7: *The space of hypergranules.*

The properties of $C_X^0(t_1|t_0)$ and $C_X^1(t_1|t_0)$ are stated in the following lemmas.

Lemma 4.1. *$C_X^0(t_1|t_0)$ satisfies the following:*

- $0 \leq C_X^0(t_1|t_0) \leq 1$.
- $C_X^0(t_1|t_0) \neq C_X^0(t_0|t_1)$ *in general.*
- $C_X^0(t_1|t_0) = 1$ *if* $t_0(X) \leq t_1(X)$.
- $C_X^0(t_1|t_0) = 0$ *if there is no* t *such that* $t_0(X) \cup t_1(X) \leq t(X)$.

Lemma 4.2. *$C_X^1(t_1|t_0)$ satisfies the following:*

- $0 \leq C_X^1(t_1|t_0) \leq 1$.
- $C_X^1(t_1|t_0) \neq C_X^1(t_0|t_1)$ *in general.*
- $C_X^1(t_1|t_0) = 1$ *if* $t_0(X) \cup t_1(X)$ *doesn't cover any* $t(X)$ *where* $t(X) \neq t_0(X)$ *and* $t(X) \neq t_1(X)$.
- $C_X^1(t_1|t_0) = 0$ *if* $t_1(X) = \emptyset$.

The proofs of the two lemmas are straightforward.

Having the above two measures, we devise the following algorithm (C2) for the assignment problem. Let $t \in V$.

- For each $s \in \bigcup_i E_i$, calculate $C_\Omega^0(s|t)$ and $C_\Omega^1(s|t)$.

- Let Q be the set of $s \in \bigcup_i E_i$ which have maximal C_X^0 values. If Q has only one element, namely $Q = \{s\}$, then label t by the label of s. Otherwise, let R be the set of $s \in Q$ which have maximal C_X^1 values. If R has only one element, namely $R = \{s\}$, then label t by the label of s. Otherwise, label t by the label of the hypergranule in R which has the highest coverage.

5 An example

In this section we are going to illustrate both the LM and C2 algorithms using one example.

Table 8: A sample of 4 rows of the Iris data

SL	SW	PL	PW	Spec.	SL	SW	PL	PW	Spec.	SL	SW	PL	PW	Spec.
50	36	14	02	Setosa	61	28	40	13	Versicolor	63	28	51	15	Virginica
54	39	17	04	Setosa	63	25	49	15	Versicolor	61	26	56	14	Virginica
46	34	14	03	Setosa	62	29	43	13	Versicolor	72	30	58	16	Virginica
50	34	15	02	Setosa	60	27	51	16	Versicolor	67	31	56	24	Virginica

LM

First of all, we illustrate LM. For the Setosa class, the sum of the first two rows results in

$$\langle\{50, 54\}, \{36, 39\}, \{14, 17\}, \{2, 4\}\rangle.$$

Since this hypergranule doesn't cover [3] any tuple in the other two classes, this hypergranule is equilabelled. It can be similarly verified that the sum of any pair of tuples in Setosa class is equilabelled. The set of all sums is shown in Table 9, and it is a cover for this class. However this cover is not an E-set as the sum of the first two hypergranules in Table 9, $\langle\{46, 50, 54\}, \{34, 36, 39\}, \{14, 17\}, \{2, 3, 4\}\rangle$, is also equilabelled. Eventually we get an E-set for Setosa class, which has only one hypergranule – $\langle\{46, 50, 54\}, \{34, 36, 39\}, \{14, 15, 17\}, \{2, 3, 4\}\rangle$. The same procedure can be applied to the other two classes. As a result, we get a reduced dataset – a set of E-sets one for each class, shown in Table 10. Note that the E-set for the Virginica class has two hypergranules as summing them would result in loss of consistency (the sum is not equilabelled).

C2

Now we illustrate C2. Consider $t \stackrel{\text{def}}{=} \langle 48, 34, 16, 2\rangle$. Following the C2 procedure, we calculate the C_0 and C_1 values for all the hypergranules as follows: $C^0(s_0|t) = 1$, $C^0(s_1|t) = 0$, $C^0(s_2|t) = 0$, $C^0(s_3|t) = 0$. There is no need to calculate C^1 values since there is only one hypergranule having the

[3] In terms of the $\leq$ ordering on page 3

Table 9: *The set of all sums of pair of Setosa tuples*

Attribute				Class
Sepal length	Sepal width	Petal length	Petal width	
$\{50, 54\}$	$\{36, 39\}$	$\{14, 17\}$	$\{2, 4\}$	Setosa
$\{50, 46\}$	$\{34, 36\}$	$\{14\}$	$\{2, 3\}$	Setosa
$\{50\}$	$\{34, 36\}$	$\{14, 15\}$	$\{2\}$	Setosa
$\{46, 54\}$	$\{34, 39\}$	$\{14, 17\}$	$\{3, 4\}$	Setosa
$\{50, 54\}$	$\{34, 39\}$	$\{15, 17\}$	$\{2, 4\}$	Setosa
$\{46, 50\}$	$\{34\}$	$\{14, 15\}$	$\{2, 3\}$	Setosa

Table 10: *Model: The set of all E-sets.*

ID	Attribute				Class
	Sepal length	Sepal width	Petal length	Petal width	
s_0	$\{46, 50, 54\}$	$\{34, 36, 39\}$	$\{14, 15, 17\}$	$\{2, 3, 4\}$	Setosa
s_1	$\{61, 62, 63\}$	$\{25, 28, 29\}$	$\{40, 43, 49\}$	$\{13, 15\}$	Virginica
s_2	$\{60\}$	$\{27\}$	$\{51\}$	$\{16\}$	Virginica
s_3	$\{61, 63, 67, 72\}$	$\{26, 28, 30, 31\}$	$\{51, 56, 58\}$	$\{14, 15, 16, 24\}$	Versicolor

maximal C^0 value. Then we can label t by the label of s_0 – Setosa. This is in fact the single coverage case.

Now we consider another tuple $t \stackrel{\text{def}}{=} \langle 58, 40, 55, 17\rangle$. The C^0 values for all hypergranules are 0. For C^1 values, we have $C^1(s_0|t) = 1$, $C^1(s_1|t) = 0$, $C^1(s_2|t) = 1$, and $C^1(s_3|t) = 0$. Since s_0 has higher coverage (4 cases covered) than s_2 (1 case covered), we label t by Setosa. This is the non-coverage case.

Using the hypergranules in Table 10 to label the whole of Iris data, we get a success rate of 88.7%.

The complete Iris data and the complete set of hypergranules found by LM are listed in Appendix.

6 Evaluation

In order to test the LM and C2 algorithm, we have used a number of datasets available from the UCI machine learning repository from where the appropriate references of origin can be obtained. Most of the datasets are frequently used in literature. Some general information about these datasets is given in Table 11.

Most of the datasets contain missing values. Missing values usually mean either that the actual values

Table 11: General information about the datasets.

Datasets	Features	Examples	Classes
Annealing	38	798	6
Australian	14	690	2
Auto	25	205	6
Breast	9	286	2
Diabetes	8	768	2
German	20	1000	2
Glass	9	214	6
Heart	13	270	2
Hepatitis	19	155	2
Horse-Colic	22	368	2
Iris	4	150	3
Sonar	60	208	2
Tic-Tac-Toe	9	958	2
Vehicle	18	846	4
Vote	18	232	2

are not important, or that the actual values are not available. Our philosophy is that, whichever case this is, missing values should not contribute in the modelling process and the classification process. As a result, we deal with missing values simply by filling them with empty set, which contribute nothing to either modelling or classification because our hypergranules contain sets of values instead of single values.

To achieve our objective, we need a standard data mining algorithm for benchmarking. We chose C4.5 (see [14]) as it is one of the most extensively used algorithms in the literature and it is widely available so that the experiment results can be easily repeated, if needed; in the present study, we have used the C4.5 module of the Clementine [**?**] package. We have used 5-fold cross validation for both C4.5 and LM.

The results are shown in Table 12. We analysed the prediction success of C4.5 and LM, and the cross-classification is a straight line. Therefore we can say that LM is comparable to C4.5.
We also analysed the relationship between reduction ratio and prediction success. The regression line is estimated by PredictionSuccess = 1.044*ReductionRatio - 0.128. This suggests that the reduction ratio is a good measure of prediction success.

Table 12: Prediction success of C4.5 and LM and the reduction ratio obtained by LM. These results are based on the new C^0 and C^1 measures. The LM results were obtained with pruning. The reduction ratio is defined as $|D| - |\bigcup_i E_i|/|D|$, where E_i is the E-set for class M_i, and D is the dataset.

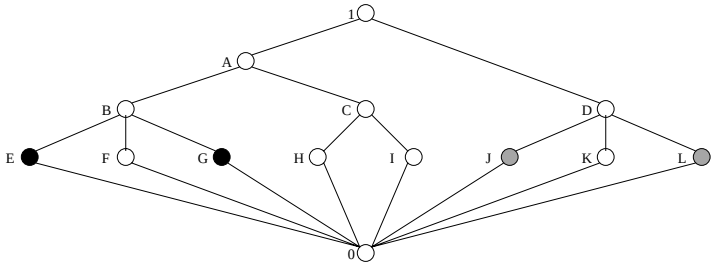
Dataset	Prediction success		Reduction ratio
	C4.5	LM	
Annealing	91.8	93.6	90.6
Australian	85.2	83.5	85.2
Auto	72.2	76.1	89.9
Breast	74.7	72.6	82.5
Diabetes	72.9	71.7	82.0
German	70.5	72.5	96.0
Glass	81.3	82.7	87.3
Heart	77.1	77.0	84.4
Hepatitis	80.7	80.0	89.4
Horse-Colic	80.9	78.2	83.0
Iris	94.0	96.0	97.6
Sonar	69.4	69.7	90.8
Tic-Tac-Toe	86.2	83.5	88.5
Vehicle	69.9	62.2	84.1
Vote	95.1	97.0	98.5
Wine	94.3	94.4	96.1
Average	79.95	79.75	88.70

7 Related work

The classic work of Mitchell on version spaces [11] is directly relevant. Mitchell viewed space of all possible concept descriptions as a lattice from the most general down to the most specific. He defined the version space as the sub-lattice that is consistent with a set of labelled examples. He defined the G-set and the S-set (the subsets of descriptions that make up the most general/most specific boundaries of the sub-lattice). This work was followed up by many others, most notably Hirsh [9, 10], who discussed 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 noises.

This line of research is concerned mainly with how to find, given a set of labelled examples (i.e., a set of positive and negative examples of a concept), the G-set and the S-set, which together represents the space of all possible consistent concepts. However, for practical tasks, we usually do not need all

Figure 3: *A labelled lattice.*



concepts, but a single one which predicts or classifies best. The question of how to find such a concept is not addressed in this line of research, as far as we know.

Each E-set found by LM is a single concept consistent with a class of labelled examples. In the case of two class classification problem, there will be one E-set for the positive case, and another for the negative case. Different E-sets represent single concepts for different classes. To justify the selection of such single concepts, consider the abstract lattice in Figure 3. In this labelled lattice, elements A and B are both equilabeled elements. But A has greater coverage of unlabeled elements than B; in other words, A is more general than B (or B is more specific than A). In the spirit of least general generalisation (LGG) [4], we should prefer B to A in our pursuit of a single concept for the dark black class. LM is designed having this in mind, which concludes that the E-set for the dark black class is $\{B\}$, and the E-set for the light black class is $\{D\}$.

Each E-set is a subset of the S-set in the following sense. Let S and T be two arbitrary sets, $d_0 \stackrel{\text{def}}{=} \langle s, t_0\rangle, d_1 \stackrel{\text{def}}{=} \langle s, t_1\rangle \in S \times T$. Let Mit be the operation implied in the S-set examples used by Mitchell ([11],page 214). If $t_0 \neq t_1$, then $Mit(d_0, d_1) = \langle s, ?\rangle$, where the question mark means that the elements in T are unimportant ([11],page 205). In the context of our decision system, this amounts to $Mit(d_0, d_1) = \langle s, T\rangle$, since for any $d' = \langle s', t'\rangle$, the comparison between d' and $Mit(d_0, d_1)$ regarding which is more general or specific will be made by comparing s and s' irrespective of t', t_0, t_1. This is to say that the Mit operation generalises d_0 and d_1 to $\langle s, T\rangle$, which clearly assumes too much extra information which may not be true in reality. However, by some misuse of notation, our approach results in $LM(d_0, d_1) = \langle s, \{t_0, t_1\}\rangle$, which uses only available information. In this sense we say that each E-set is a subset of the S-set. No decision rules is provided for classification in [11], nor has one been found elsewhere in this line of research, as far as we know. Furthermore, no application in a practical setting has been found.

[4] LGG says that if two clauses c_1 and c_2 are true, it is very likely that their most specific generalisation will also be true [13, 3].

Mitchell used an example to illustrate his idea. The training set is as follows:

$$\{(\text{Large Red Triangle})(\text{Small Blue Circle})\} : + \tag{7.1}$$

$$\{(\text{Large Blue Circle})(\text{Small Red Triangle})\} : + \tag{7.2}$$

$$\{(\text{Large Blue Triangle})(\text{Small Blue Triangle})\} : - \tag{7.3}$$

Each line describes a pair of *unordered* objects, which is labelled as either positive (+) or negative (-). The S-set and G-set obtained are

$$S-set: \quad [\langle(?\ \text{Red Triangle})(?\ \text{Blue Circle})\rangle, \langle(\text{Large}\ ?\ ?)(\text{Small}\ ?\ ?)\rangle]$$

$$G-set: \quad [\langle(?\ \text{Red}\ ?)(?\ ?\ ?)\rangle, \langle(?\ ?\ \text{Circle})(?\ ?\ ?)\rangle]$$

This example can be turned into a set of 8 decision systems as shown in Table 13, and the E-sets corresponding to these tables are shown in Table 14. The collection of different incomparable E-sets for the positive class is $\{\langle\{L,S\},R,T\rangle, \langle\{L,S\},B,C\rangle, \langle L,\{B,R\},\{C,T\}\rangle, \langle S,\{B,R\},\{C,T\}\rangle\}$. Clearly this is exactly the S-set except that $\{L,S\},\{B,R\},\{C,T\}$ are replaced by the question mark respectively. As we argued earlier, Mitchell's results generalise beyond given information.

In sum, Mitchell's version space is where the underlying concepts should belong to, though for larger problems the space could be large, and the use of the space rest with the users. Our approach attempts to find a single concept which is relatively conservative in the sense of *least general generalisation* principle.

8 Summary and conclusion

In this paper we have presented a novel approach to the problem of classificatory filtering – preserving classification information in the process of data reduction. Our approach is a generalisation of the filtering method discussed in [2].

We presented an algorithm (LM) in the context of lattice. In the context of decision systems, we look at hypergranules and it turns out that the collection of all hypergranules in a given domain is a lattice. LM works, in decision systems, by collecting attribute values while preserving classification information. It inputs a decision system, and outputs a set of maximal hypergranules that, collectively, is consistent with the original decision system but is much smaller in size.

We also discussed the problem of assigning classification labels to new data based on filtered data, with respect to three cases: single coverage, mutiple coverage, and non coverage. We proposed and

Table 13: Representing Mitchell's example using multiple tables.

U	Size	Colour	Shape	Class
u_0	L	R	T	+
u_1	L	B	C	+
u_2	L	B	T	-

U	Size	Colour	Shape	Class
u_0	L	R	T	+
u_1	L	B	C	+
u_2	S	B	T	-

U	Size	Colour	Shape	Class
u_0	L	R	T	+
u_1	S	R	T	+
u_2	L	B	T	-

U	Size	Colour	Shape	Class
u_0	L	R	T	+
u_1	S	R	T	+
u_2	S	B	T	-

U	Size	Colour	Shape	Class
u_0	S	B	C	+
u_1	L	B	C	+
u_2	L	B	T	-

U	Size	Colour	Shape	Class
u_0	S	B	C	+
u_1	L	B	C	+
u_2	S	B	T	-

U	Size	Colour	Shape	Class
u_0	S	B	C	+
u_1	S	R	T	+
u_2	L	B	T	-

U	Size	Colour	Shape	Class
u_0	S	B	C	+
u_1	S	R	T	+
u_2	S	B	T	-

Table 14: E-sets corresponding to the tables in Table 13.

U	Size	Colour	Shape	Class
u'_0	L	R	T	+
u'_1	L	B	C	+
u'_2	L	B	T	-

U	Size	Colour	Shape	Class
u'_0	L	$\{B, R\}$	$\{C, T\}$	+
u'_1	S	B	T	-

U	Size	Colour	Shape	Class
u'_0	$\{L, S\}$	R	T	+
u'_1	L	B	T	-

U	Size	Colour	Shape	Class
u'_0	$\{L, S\}$	R	T	+
u'_1	S	B	T	-

U	Size	Colour	Shape	Class
u'_0	$\{L, S\}$	B	C	+
u'_1	L	B	T	-

U	Size	Colour	Shape	Class
u'_0	$\{L, S\}$	B	C	+
u'_1	S	B	T	-

U	Size	Colour	Shape	Class
u'_0	S	$\{B, R\}$	$\{C, T\}$	+
u'_1	L	B	T	-

U	Size	Colour	Shape	Class
u'_0	S	B	C	+
u'_1	S	R	T	+
u'_2	S	B	T	-

justified two measures: one measures the likelihood that one hypergranule covers another; another measures the likelihood that merging two hypergranules covers others. Based on the two measures we devised an algorithm (C2) which efficiently classifies new data.

We evaluated both LM and C2 using some real world datasets. We used LM to discover "knowledge" – hypergranules from these datasets, and then we used C2 to classify new data. We used 5-fold cross validation method for the evaluatation and we found that the result is comparable to that of C4.5. Analysis of the result by statistical test methods showed that there is no statistical difference between LM/C2 and C4.5. Further regression analysis shows that the reduction ratio is a strong indicator of prediction success.

A Iris data

Table 15: Iris data

SL	SW	PL	PW	Spec.	SL	SW	PL	PW	Spec.	SL	SW	PL	PW	Spec.
51	35	14	02	Setosa	70	32	47	14	Versicolor	63	33	60	25	Virginica
54	39	17	04	Setosa	57	28	45	13	Versicolor	76	30	66	21	Virginica
46	34	14	03	Setosa	63	33	47	16	Versicolor	49	25	45	17	Virginica
50	34	15	02	Setosa	49	24	33	10	Versicolor	73	29	63	18	Virginica
44	29	14	02	Setosa	66	29	46	13	Versicolor	67	25	58	18	Virginica
49	30	14	02	Setosa	64	32	45	15	Versicolor	58	27	51	19	Virginica
47	32	13	02	Setosa	69	31	49	15	Versicolor	71	30	59	21	Virginica
46	31	15	02	Setosa	55	23	40	13	Versicolor	63	29	56	18	Virginica
50	36	14	02	Setosa	65	28	46	15	Versicolor	65	30	58	22	Virginica
49	31	15	01	Setosa	52	27	39	14	Versicolor	72	36	61	25	Virginica
54	37	15	02	Setosa	50	20	35	10	Versicolor	65	32	51	20	Virginica
48	34	16	02	Setosa	59	30	42	15	Versicolor	64	27	53	19	Virginica
48	30	14	01	Setosa	60	22	40	10	Versicolor	68	30	55	21	Virginica
43	30	11	01	Setosa	61	29	47	14	Versicolor	57	25	50	20	Virginica
58	40	12	02	Setosa	56	29	36	13	Versicolor	58	28	51	24	Virginica
57	44	15	04	Setosa	67	31	44	14	Versicolor	64	32	53	23	Virginica
54	39	13	04	Setosa	56	30	45	15	Versicolor	65	30	55	18	Virginica
51	35	14	03	Setosa	58	27	41	10	Versicolor	77	38	67	22	Virginica
57	38	17	03	Setosa	62	22	45	15	Versicolor	77	26	69	23	Virginica
51	38	15	03	Setosa	56	25	39	11	Versicolor	60	22	50	15	Virginica
54	34	17	02	Setosa	59	32	48	18	Versicolor	69	32	57	23	Virginica
51	37	15	04	Setosa	61	28	40	13	Versicolor	56	28	49	20	Virginica
46	36	10	02	Setosa	63	25	49	15	Versicolor	77	28	67	20	Virginica
51	33	17	05	Setosa	61	28	47	12	Versicolor	63	27	49	18	Virginica
48	34	19	02	Setosa	64	29	43	13	Versicolor	67	33	57	21	Virginica
50	30	16	02	Setosa	66	30	44	14	Versicolor	72	32	60	18	Virginica
50	34	16	04	Setosa	68	28	48	14	Versicolor	62	28	48	18	Virginica
52	35	15	02	Setosa	67	30	50	17	Versicolor	61	30	49	18	Virginica
52	34	14	02	Setosa	60	29	45	15	Versicolor	64	28	56	21	Virginica
47	32	16	02	Setosa	57	26	35	10	Versicolor	72	30	58	16	Virginica
48	31	16	02	Setosa	55	24	38	11	Versicolor	74	28	61	19	Virginica
54	34	15	04	Setosa	55	24	37	10	Versicolor	79	38	64	20	Virginica
52	41	15	01	Setosa	58	27	39	12	Versicolor	64	28	56	22	Virginica
55	42	14	02	Setosa	60	27	51	16	Versicolor	63	28	51	15	Virginica
49	31	15	01	Setosa	54	30	45	15	Versicolor	61	26	56	14	Virginica
50	32	12	02	Setosa	60	34	45	16	Versicolor	77	30	61	23	Virginica
55	35	13	02	Setosa	67	31	47	15	Versicolor	63	34	56	24	Virginica
49	31	15	01	Setosa	63	23	44	13	Versicolor	64	31	55	18	Virginica

Cont. from previous page

SL	SW	PL	PW	Spec.	SL	SW	PL	PW	Spec.	SL	SW	PL	PW	Spec.
44	30	13	02	Setosa	56	30	41	13	Versicolor	60	30	48	18	Virginica
51	34	15	02	Setosa	55	25	40	13	Versicolor	69	31	54	21	Virginica
50	35	13	03	Setosa	55	26	44	12	Versicolor	67	31	56	24	Virginica
45	23	13	03	Setosa	61	30	46	14	Versicolor	69	31	51	23	Virginica
44	32	13	02	Setosa	58	26	40	12	Versicolor	58	27	51	19	Virginica
50	35	16	06	Setosa	50	23	33	10	Versicolor	68	32	59	23	Virginica
51	38	19	04	Setosa	56	27	42	13	Versicolor	67	33	57	25	Virginica
48	30	14	03	Setosa	57	30	42	12	Versicolor	67	30	52	23	Virginica
51	38	16	02	Setosa	57	29	42	13	Versicolor	63	25	50	19	Virginica
46	32	14	02	Setosa	62	29	43	13	Versicolor	65	30	52	20	Virginica
53	37	15	02	Setosa	51	25	30	11	Versicolor	62	34	54	23	Virginica
50	33	14	02	Setosa	57	28	41	13	Versicolor	59	30	51	18	Virginica

The hypergranules obtained by the LM – algorithm for the Iris data are given in Table 16.

Table 16: Hypergranules for Iris

Attribute							Class
Sepal length		Sepal width		Petal length		Petal width	
$\{43, \ldots, 58\}$	$\times$	$\{23, 29, \ldots, 44\}$	$\times$	$\{10, \ldots, 17, 19\}$	$\times$	$\{1, \ldots, 6\}$	Setosa
$\{49, \ldots 52, 54, \ldots, 70\}$	$\times$	$\{20, 22, \ldots, 34\}$	$\times$	$\{30, 33, 35, \ldots, 49\}$	$\times$	$\{10, \ldots, 16\}$	Versicolor
$\{56, \ldots, 59, 61, \ldots, 69, 71, \ldots, 74, 76, 77, 79\}$	$\times$	$\{25, \ldots, 34, 36, 38\}$	$\times$	$\{49, \ldots, 61, 63, 64, 66, 67, 69\}$	$\times$	$\{18, \ldots, 25\}$	Virginica
$\{61, 63\}$	$\times$	$\{26, 28\}$	$\times$	$\{51, 56\}$	$\times$	$\{14, 15\}$	Virginica
$\{60\}$	$\times$	$\{30\}$	$\times$	$\{48\}$	$\times$	$\{18\}$	Virginica
$\{72\}$	$\times$	$\{30\}$	$\times$	$\{58\}$	$\times$	$\{16\}$	Virginica
$\{62\}$	$\times$	$\{28\}$	$\times$	$\{48\}$	$\times$	$\{18\}$	Virginica
$\{60\}$	$\times$	$\{22\}$	$\times$	$\{50\}$	$\times$	$\{15\}$	Virginica
$\{49\}$	$\times$	$\{25\}$	$\times$	$\{45\}$	$\times$	$\{17\}$	Virginica
$\{60\}$	$\times$	$\{27\}$	$\times$	$\{51\}$	$\times$	$\{16\}$	Versicolor
$\{67\}$	$\times$	$\{30\}$	$\times$	$\{50\}$	$\times$	$\{17\}$	Versicolor
$\{59\}$	$\times$	$\{32\}$	$\times$	$\{48\}$	$\times$	$\{18\}$	Versicolor

References

[1] Ivo Düntsch and Günther Gediga. Statistical evaluation of rough set dependency analysis. *International Journal of Human–Computer Studies*, 46:589–604, 1997.

[2] Ivo Düntsch and Günther Gediga. Simple data filtering in rough set systems. *International Journal of Approximate Reasoning*, 18(1–2):93–106, 1998.

[3] Saso Dzeroski. *Inductive Logic Programming and Knowledge Discovery in Databases*, pages 117–152. AAAI Press / The MIT Press, 1996.

[4] Usama M. Fayyad. Editorial. *Data Mining and Knowledge Discovery – An International Journal*, 1(3), 1997.

[5] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Ann. Eugen.*, 7:179–188, 1936.

[6] G. Gigerenzer. *Messung und Modellbildung in der Phychologie*. Birkhäuser, Basel, 1981.

[7] George Grätzer. *General Lattice Theory*. Birkhäuser, Basel, 1978.

[8] J. Guan and D. A. Bell. *Evidence Theory and Its Applications, Volume 1*. Elsevier, The Netherlands, 1991. Studies in Computer Science and Artificial Intelligence 7.

[9] H. Hirsh. Incremental version-space merging. In *Machine Learning: Proc. of 7th International Conference*, pages 330–338, 1990.

[10] H. Hirsh. Learning from data with bounded inconsistency. In *Machine Learning: Proc. of 7th International Conference*, pages 32–39, 1990.

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

[12] Ewa Orlowska. Logic of nondeterministic information. *Studia Logica*, 44:93–102, 1985.

[13] G. Plotkin. *A Note on Inductive Generalization*, pages 153–163. Edinburgh University Press, Edinburgh, UK, 1969.

[14] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993.

[15] Sholom M. Weiss and Nitin Indurkhya. *Predictive Data Mining: A Practical Guide*. Morgan Kaufmann Publishers, Inc., 1997.