ELEKTROTEHNIŠKI VESTNIK

ELECTROTECHNICAL REVIEW
LJUBLJANA - SLOVENIA

1992

ELVEA2, vol. 59, št. 5, str. 233–320
Ljubljana, december 1992

5

Visualization in machine learning

Dunja Mladenić, Franc Solina, Ljubljana

UDK 681.3

Manuscript received 1992–12–05

Key words: inductive concept learning, visualization, parallel coordinates, real-world problem

Ključne besede: induktivno učenje konceptov, vizualizacija, paralelne koordinate, realni problem

Vizualizacija v avtomatskem učenju

V članku opisujemo metodo za vizualizacijo podatkov v avtomatskem učenju. Metoda sloni na metodi paralelnih koordinat, katera je originalno razvita za vizualizacijo v večdimenzionalni geometriji. Opisana metoda je implementirana v sistemu za učenje ATRIS in omogoča vizualizacijo vhodnih podatkov (n-teric vrednosti atributov) in naučenih opisov koncepta (če-potem pravil). Rezultati poskusov na realnem problemu, prikazujejo uporabnost metode v analizi problemov avtomatskega učenja.

The paper presents a method for visualizing data in machine learning. The method is based on the parallel coordinates method which was previously used for visualizing multidimensional geometry. The method, implemented in a rule learning system ATRIS, enables visualization of domain data (attribute-value tuples) and concept descriptions (if-then rules). Results of the experiments on a real-world domain illustrate the usefulness of the method for analysing machine learning problems.

1 Introduction

Learning concepts from examples, also called inductive concept learning, is the most common and best understood form of machine learning. The task of inductive concept learning is to find a concept description for a given set of examples. The set of examples contains empirical knowledge collected by experts. Induced concept description generalize properties of examples. When analysing concept learning problems it is most useful to have a means of representing data and induced concept descriptions also in a graphical form.

Visualization of data, concepts and relations is not limited only to physical entities but is important also for understanding of abstract entities. Graphical representation of such entities can relay to users a lot of information in a very short time. Visualization enables easier and faster understanding of complex problems. This is supported by the current trend of visualization in computer science. While in some application areas natural or traditional ways of graphically representing the problem area exist new ways must be found for other areas.

A visualizing method in machine learning aims at representing the space of examples (attribute-value tuples) and the corresponding concept descriptions (induced if-than rules) which are both multidimensional with a dimension depending on the number of attributes. In machine learning, an early approach to data visualization was by using a simple table representation of examples and concept descriptions [6]. Other useful methods for data analysis exist; some of them are concerned with the visualization of only two dimensions at once [1] while the others are suitable for multidimensional analysis [4]. Most of them have a limited number of dimensions they can deal with.

To overcome this limitation, the representation of points in parallel coordinates can be used. The idea for parallel coordinates based visualizing method is borrowed from Inselberg and Dimsdale [5]. They used the parallel coordinates method for representing multidimensional points and, furthermore, for the visualization of multidimensional geometry. The paper presents the adaptation of this method to machine learning problems, which enables the analysis of data (examples) and induced concept descriptions (hypotheses). For this purpose, only the representation of individual points in the example and hypotheses space with parallel coordinates is used.

The use of the adapted parallel coordinates visualizing method is described on a simple machine learning domain [8]. To illustrate the usefulness of this method some results on a real-world medical domain are presented as well.

The second section describes the table visualizing method [6] on a simple machine learning domain from [8]. In the third section, parallel coordinates visualizing method is described and its use illustrated on the same simple domain. Its use on a real-world medical domain is shown and discussed in the fourth section. The paper concludes with a summary of our approach and some ideas for further work.

2 Visualization with tables

2.1 *The table visualizing method*

In the table visualizing method [6], the points in N-dimensional space are represented with a two-dimensional

Reviewed 1992–12–15

table. To each dimension, a row (or a column) at the margins of the table is assigned for the appropriate enumeration of its possible values. For example, if we have a three-dimensional space, first a row is assigned to one dimension, a column is assigned to the second dimension, and to the third dimension a row above the last assigned row is assigned. Suppose we have a problem described in three dimensions d1, d2 and d3 and that they have three, two and two values each, respectively. The table used to visualize this problem is given in Table 1. For example, the point **X** in the table represents the point (d1v2, d2v2, d3v2) in the d1 × d2 × d3 space. Note that in this table elements in the same row should have the same value of the second dimension.

d3	d3v1	d3v2	d3v1	d3v2	d3v1	d3v2	
d1	d1v1		d1v2		d1v3		d2
							d2v1
				X			d2v2

Table 1. Table visualizing method.

When assigning rows and columns to dimensions, it is advisable to sort dimensions according to the number of values. Rows and columns near the table margins are chosen for dimensions with more values.

2.2 *An example problem domain*

To describe the visualizing methods a simple 'Saturday morning' [8] domain is used. In this machine learning domain, 14 training examples are given (see Table 2). Each example is labeled with its class: suitable (no, yes), saying if a morning is suitable for some unspecified activity, and 4 attributes: outlook (sunny, overcast, rain), temperature (hot, mild, cool), humidity (high, normal) and windy (no, yes), describing a Saturday morning.

class	attributes			
suitable	outlook	temperature	humidity	windy
no	sunny	hot	high	no
no	sunny	hot	high	yes
yes	overcast	hot	high	no
yes	rain	mild	high	no
yes	rain	cool	normal	no
no	rain	cool	normal	yes
yes	overcast	cool	normal	yes
no	sunny	mild	high	no
yes	sunny	cool	normal	no
yes	rain	mild	normal	no
yes	sunny	mild	normal	yes
yes	overcast	mild	high	yes
yes	overcast	hot	normal	no
no	rain	mild	high	yes

Table 2. Examples for the 'Saturday morning' domain.

A machine learning algorithm such as ATRIS, can be used to induce if-then rules from the given set of examples. Two rules for the no-class and three for the yes-class are given below.

Rule 1: **if** outlook = sunny and humidity = high **than** suitable = no

Rule 2: **if** outlook = rain and windy = yes **than** suitable = no

Rule 3: **if** outlook = sunny and humidity = normal **than** suitable = yes

Rule 4: **if** outlook = overcast **than** suitable = yes

Rule 5: **if** outlook = rainy and windy = no **than** suitable = yes

2.3 *Visualizing examples and rules with tables*

The table visualizing method was already applied to machine learning by Michalski and Stepp [6]. To apply this method on machine learning data, attributes are viewed as dimensions and examples as points in this attribute space (in the same way as in [6]). To each of the class values a different symbol is assigned (− to the no-class, + to the yes-class), in this way all the examples can be shown in the same Table 3. For each class, a set of rules can be shown on the table with examples (Tables 4 and 5) in order to illustrate how the rules cover the examples. E.g. in Table 4 the first column filled with (———) visualized the above given Rule 1.

3 Visualization with parallel coordinates

3.1 *The parallel coordinates method*

The parallel coordinates method maps $R^N \to R^2$, so that the relation among N variables is represented by its planar image. In the two-dimensional (xy-Cartesian coordinate) space, N real lines, labeled with $x_1, x_2, \ldots, x_N$, are placed equidistant and parallel to the y-axis. These lines are axes of the parallel coordinate system. A point C with coordinates $(c_1, c_2, \ldots, c_N)$ is represented by a polygonal line, whose N vertices are at c_i on the x_i-axis for $i = 1 \ldots N$ (see Figure 1). A one-to-one correspondence between points in R^N and polygonal lines in R^2 with vertices on the x_1, x_2, …, x_N axes is established.

Starting from the point in R^N, the method can be used for multidimensional geometrical objects visualization [5]. In this paper, the parallel coordinates method is used only for the representation of points.

3.2 *Visualizing examples and rules with parallel coordinates*

Like in the simple table visualizing method from Section 2.3, attributes are viewed as dimensions and examples as points in the so-formed attribute space. Attributes are enumerated and to each attribute an x-axis is assigned. Attribute values are also enumerated and an interval* on the appropriate x-axis is assigned to each of them. Ordering of attributes and attribute values must be fixed at the beginning of the visualization since different orderings result with different pictures. To produce different, maybe even better visualization, ordering of values (only for discrete, nominal attributes) could be changed. The distance between x-axes and the interval length are calculated from the given dimensions of the visualizing window (part of or the whole screen).

*Rather than a point, an interval is assigned also to discrete attributes to enable the separation between the individual examples.

humidity	high	normal	high	normal	high	normal		
outlook	sunny		overcast		rain		temp.	windy
				+		−	cool	yes
		+				+		no
		+	+		−		mild	yes
	−				+	+		no
	−						hot	yes
	−		+	+				no

Table 3. Table visualization of examples for the 'Saturday morning' domain.

humidity	high	normal	high	normal	high	normal		
outlook	sunny		overcast		rain		temp.	windy
	———			+	———	—−—	cool	yes
	———	+				+		no
	———	+	+		—−—	———	mild	yes
	—−—				+	+		no
	—−—				———	———	hot	yes
	—−—		+	+				no

Table 4. Table visualization of no-class rules for the 'Saturday morning' domain. All no-class examples (−) are covered by the rules (———).

humidity	high	normal	high	normal	high	normal		
outlook	sunny		overcast		rain		temp.	windy
		———	———	—+—		−	cool	yes
		—+—	———	———	———	—+—		no
		—+—	—+—	———	−		mild	yes
	−	———	———	———	—+—	—+—		no
	−	———	———	———			hot	yes
	−	———	—+—	—+—	———	———		no

Table 5. Table visualization of yes-class rules for the 'Saturday morning' domain. All yes-class examples (+) are covered by the rules (———).

A set of examples is divided into subsets according to the class value and each subset is visualized separately.† Each example is represented by a polygonal line connecting the corresponding attribute values on the parallel coordinates. For discrete attribute the actual coordinate of the attribute value is chosen randomly from the designated attribute value interval, since for continous attribute a mapping of attribute values to designed interval is established.

In Figure 2 all examples for the 'Saturday morning' domain are visualized. On the left are the `C1` (no-class) examples, on the right the `C2` (yes-class) examples. Attribute `A1` (outlook) has values: `1` (sunny), `2` (overcast), `3` (rain); attribute `A2` (temperature) has values: `1` (hot), `2` (mild), `3` (rain); attribute `A3` (humidity) has values `1` (high), `2` (normal); attribute `A4` (windy) has values `1` (no), `2` (yes).

In Figures 3, 4 and 5 the induced rules are visualized in the same order as given in the domain description in Section 2.2. Each rule is represented separately on the same coordinates as the examples, and is visualized as the area between the polygonal lines connecting the allowable attribute values. If a particular attribute is not included in a rule, all possible values are allowed (the most general rule is visualized as the rectangle between the first and the last value of all attributes). In order to illustrate how a rule covers examples each class rule is shown individual on the same coordinates as the training examples belonging to this class. Example is covered by a rule if the whole polygonal line representing this example, lies inside the contour of the rule (grey area). The machine learning system induces a set of rules for each class. When inducing rules for a given class, the system tries to cover the examples from this class and to avoid the examples from other classes. The union of all rules for a given class should cover all examples from this class. Exact covering criterion might be replaced by some other non-exact criterion in order to enable handling of noise in data. If some examples remain uncovered they might be considered as outliers or noise.

†Alternatively, it is possible to visualize all the examples on the same coordinate system, and separate the classes by colours or patterns.

4 Parallel coordinates method in problem analysis

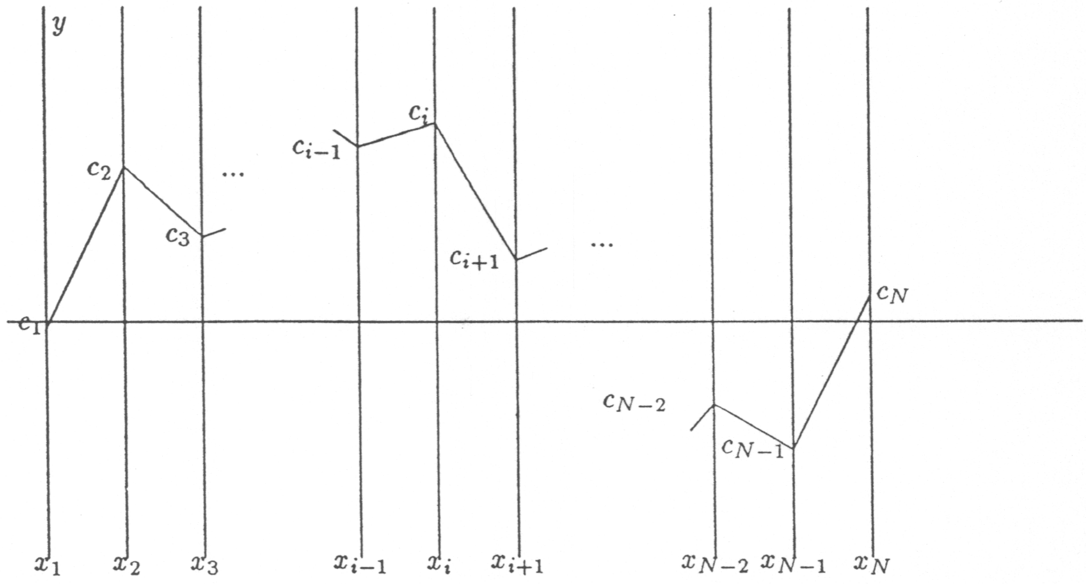

Figure 1. Visualizing a single R^N point with parallel coordinates method.

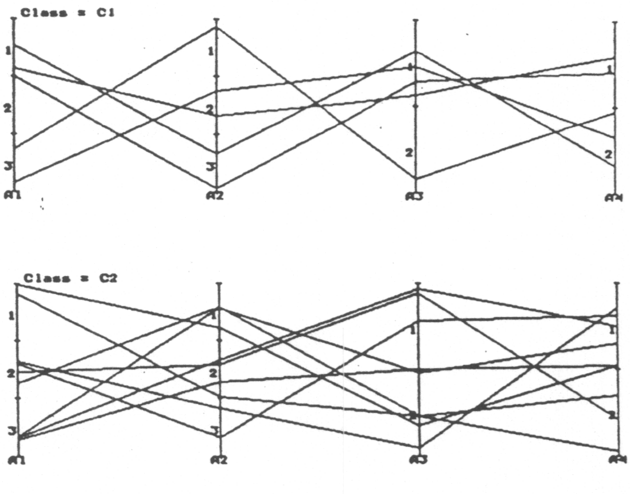

Figure 2. Examples for the 'Saturday morning' domain.

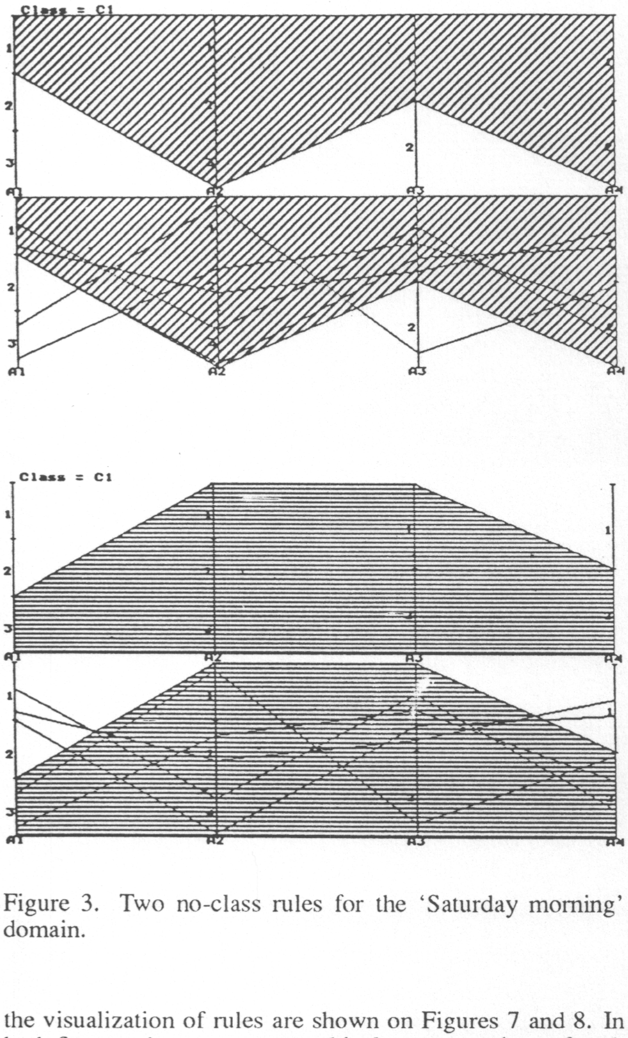

Figure 3. Two no-class rules for the 'Saturday morning' domain.

4.1 Real-world problem

The parallel coordinates method was tested on a real-world medical diagnostic domain 'lymphography' which is one of the standard testing domains for machine learning systems (e.g., Cestnik, Kononenko and Bratko 1987 [2], Clark and Niblett 1991 [3]). The domain consists of 148 examples given by values of 18 attributes, each labeled by one of 4 classes. For rule induction the machine learning system ATRIS [7] was used.

The set of examples is visualized separately for each class (for illustration only the first two classes are shown in Figure 6). When there is a large number of examples in a single class (like in class 2), it is hard to separate between the individual examples. In this case we see the contour of the polygon covered by the examples. (We have to keep in mind that the polygon is a visualization of a multidimensional space.) The rules for the corresponding class tend to adjust to the contour of the corresponding example polygon and avoid examples from the other classes. Some results of the visualization of rules are shown on Figures 7 and 8. In both figures, there are two graphical representations of each rule: without (above) and with (below) the visualization of the corresponding training examples. Visualization of examples and rules in Figures 7 and 8 enable the analysis of

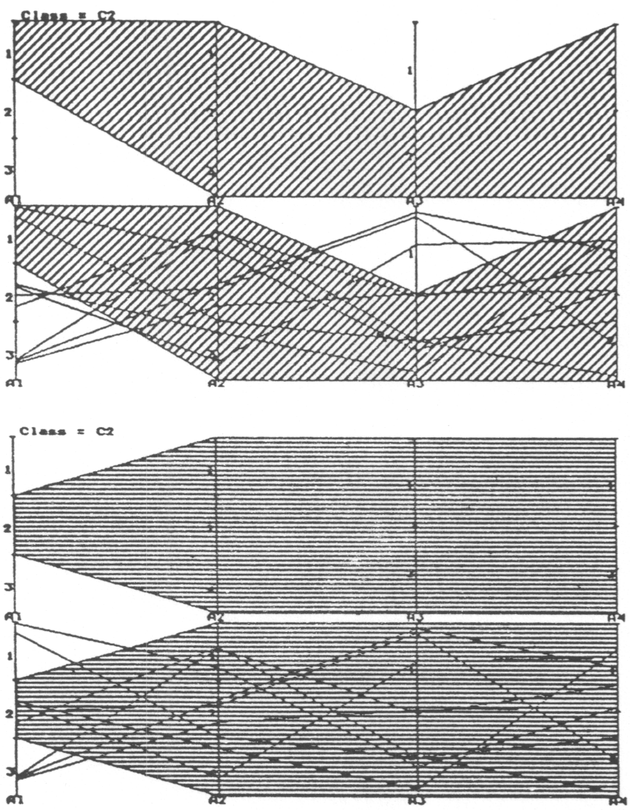

Figure 4. Two yes-class rules for the 'Saturday morning' domain.

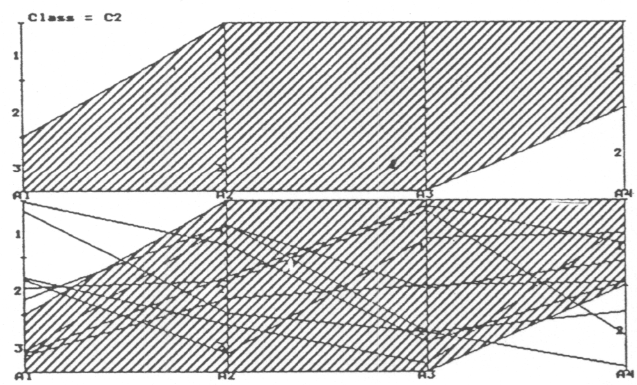

Figure 5. The third yes-class rule for the 'Saturday morning' domain. It is obvious that this rule covers examples with those values of the attribute A1 (valu 2), that were not covered by the first two rules.

the examples and the induced rules. As can be seen from Figure 7, the induced rule for class 1 covers all the examples. In Figure 8 are given just 2 out of 8 rules induced for class 2 and, therefore, on this figure many examples are not covered.

4.2 Discussion

The visualizing method can be an efficient help to a knowledge engineer using machine learning tools. First, it gives

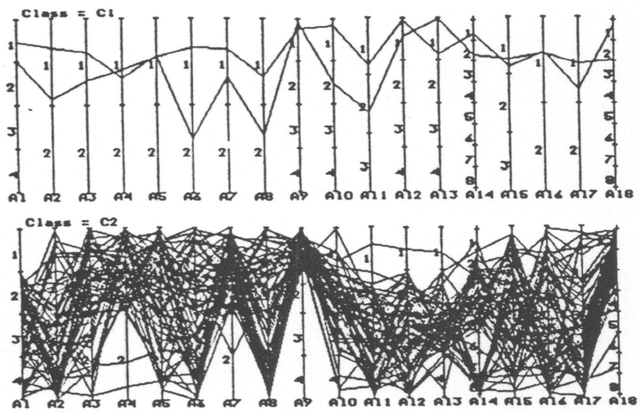

Figure 6. Examples for the 'lymphography' domain for class 1 and class 2.

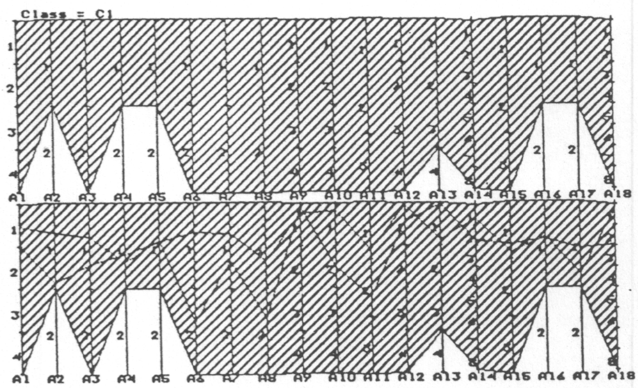

Figure 7. A rule for the first class in the 'lymphography' domain. From the figure it becomes obvious that a single rule for this class is sufficient since the rule-polygon covers all the examples in class 1.

the user a better understanding of the domain by graphically representing the training examples. Furthermore, in rule construction, machine learning systems can be 'tuned' to the given problem by appropriately selecting the parameters which influence the generality of induced rules and the handling of noisy data. In parameter tuning, the graphical representation of rules which shows the coverage of training examples is most useful. In Figure 6, some examples in class 2 are outliers in the set of examples (two examples because of the second value of the attribute A4 and one example because of the first value for attributes A11, A12, A13). When tuning the system, the rules induced for class 2 are not supposed to cover those values (these three examples are considered to be noise).

5 Conclusion

Visualization of examples and rules with the parallel coordinates method enables the analysis and better understanding of the space of examples and induced rules. The described visualizing method is connected to the machine learning system ATRIS [7] and is currently postprocessing the results of the machine learning algorithm. However, on-line visualization (tracking to gain insight) [9] during the actual learning phase could be developed on the same principles. The domain (the number of attributes and val-

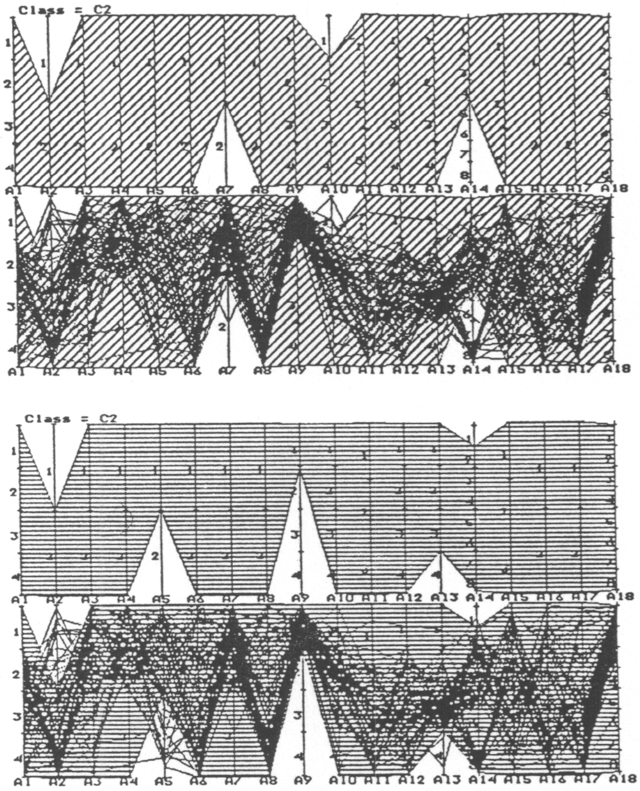

Figure 8. Two (out of 8) rules for the second class in the 'lymphography' domain. There are some uncovered rules in the figure that shows unsufficiensy of two rules for covering all the examples from class 2.

ues) is limited only with the dimensions of the window used for visualization. The method alone does not limit the dimensions of the domain.

The parallel coordinates visualizing method can be used for the analysis of induced rules and both of training and testing examples. It can also be used for visualizing rules during the process of learning (tracking) which can give useful information to the developer of a machine learning system and also to the knowledge engineer using the system.

In further work, additional use of the implemented method can be envisaged such as for visualizing the classification of new examples or for visualizing the performance of a covering algorithm by dynamically modifying the training set and the currently induced rules, deleting the examples covered by the previous rules. Introducing different colors to visualization of examples* can expose some additional information, like the difference between covered and uncovered examples.

6 Acknowledgements

This work was supported by the Slovenian Ministry of Research and Technology. Medical data set was obtained from University Medical Center in Ljubljana, and was provided by Milan Sokolič. We wish to thank Nada Lavrač for her comments to this paper.

*At the moment examples are visualized with black polygonal lines and rules with different color areas (in this paper, unfortunately they all look like grey areas).

7 References

[1] Becker, R. A., Cleveland, W. S. and Wilks, A. R. (1987) Dynamic graphics for data analysis. *Statistical Science 4*, Vol. 2, pp. 355–395.

[2] Cestnik, B., Kononenko, I. and Bratko, I. (1987) ASSISTANT 86: A knowledge elicitation tool for sophisticated users. In: Bratko, I. and Lavrač, N. (eds.) *Progress in machine learning*. Wilmslow: Sigma Press.

[3] Clark, P. and Niblett, T. (1989) The CN2 induction algorithm. *Machine Learning 1* (3), 261–284. Kluwer Academic Publishers.

[4] Dillon, W. R. and Goldstein, M. Graphical methods. In *Multivariate Analysis: Methods and applications*, pp. 191–202. John Wiley & Sons, New York.

[5] Inselberg, A. and Dimsdale, B. (1987) Parallel coordinates for visualizing multi-dimensional geometry. *Fifth International Computer Graphics Conference*. Japan.

[6] Michalski, R. S. and Stepp, R. E. (1983) Learning from observation: Conceptual clustering. In *Machine Learning, Artificial intelligence approach*, Volume I, pp. 331–363. Tioga Publishing Company, Palo Alto, California.

[7] Mladenić, D., Zupanič, D., Grobelnik, M. and Lavrač, N. (1992) Stochastic search in inductive concept learning. *Technical report CW140*, Department of Computer Science, K. U. Leuven, Belgium.

[8] Quinlan, J. R. (1986) Induction of decision trees. *Machine Learning*, Vol. 1, pp. 81–106.

[9] Rosenblum, L. J. (1989) Scientific Visualization at Research Laboratories. *Computer*, Vol. 22, No. 8, pp. 68–70.

Address of authors: Dunja **Mladenić**, dipl. ing., Jožef Stefan Institute, Computer Science Department, Artificial Intelligence Laboratory, Jamova 39, 61000 Ljubljana, Slovenia, E-mail: dunja.mladenic@ijs.si, prof. dr. Franc **Solina**, dipl. ing., University of Ljubljana, Faculty of Electrical Engineering and Computer Science, Computer Vision Laboratory, Tržaška 25, 61000 Ljubljana, Slovenia, E-mail: franc. solina@fer.uni-lj.si