R U T C O R
R ESEARCH
R E P O R T

PATTERN-BASED CLUSTERING AND ATTRIBUTE ANALYSIS

Gabriela Alexe[a] Sorin Alexe[b] Peter L. Hammer[c]

RRR-10-2003 MARCH 2003

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/~rrr

[a] RUTCOR, Rutgers University, Piscataway, NJ 08854, email: alexe@rutcor.rutgers.edu
[b] RUTCOR, Rutgers University, Piscataway, NJ 08854, email: salexe@rutcor.rutgers.edu
[c] RUTCOR, Rutgers University, Piscataway, NJ 08854, email: hammer@rutcor.rutgers.edu

RUTCOR RESEARCH REPORT

RRR-10-2003 MARCH 2003

PATTERN-BASED CLUSTERING AND ATTRIBUTE ANALYSIS

Gabriela Alexe Sorin Alexe Peter L. Hammer

Abstract. The *Logical Analysis of Data* (*LAD*) is a combinatorics, optimization and logic based methodology for the analysis of datasets with binary or numerical input variables, and binary outcomes. It has been established in previous studies that *LAD* provides a competitive classification tool comparable in efficiency with the top classification techniques available. The goal of this paper is to show that the methodology of *LAD* can be useful in the discovery of new classes of observations and in the analysis of attributes. After a brief description of the main concepts of *LAD*, two efficient combinatorial algorithms are described for the generation of all prime, respectively all spanned, patterns (rules) satisfying certain conditions. It is shown that the application of classic clustering techniques to the set of observations represented in prime pattern space leads to the identification of a subclass of, say positive, observations, which is accurately recognizable, and is sharply distinct from the observations in the opposite, say negative, class. It is also shown that the set of all spanned patterns allows the introduction of a measure of significance and of a concept of monotonicity in the set of attributes.

Acknowledgements: The partial support provided by ONR grant N00014-92-J-1375 and DIMACS is gratefully acknowledged.

1. Introduction and Basic Concepts

The extraction of knowledge hidden in records of past observations is a common problem in practically every area of science, engineering and business, and is the central object of study of classical disciplines, like statistics, and newer ones, like machine learning and data mining. Numerous methods have been developed to address this type of problems, and substantial success with their use is reported in the literature. One of the basic questions addressed by all the above mentioned areas is that of classification, e.g., that of recognizing – on the basis of similarities with already known observations in a given dataset Ω – whether a not-yet-seen observation (i.e., one not belonging to Ω) is positive or negative (or true or false, or sick or healthy, etc). We plan to show in this paper that the tools developed for classification can be successfully applied also for the extraction of other important information from datasets.

In order to clarify the concepts and the methods proposed in this paper we shall illustrate them on the *Breast Cancer Wisconsin* (**bcw**) dataset – one of the most frequently quoted, publicly available datasets (`http://www1.ics.uci.edu/~mlearn/MLRepository.html`). This dataset includes 699 observations, of which we shall only use those 683 for which the values of all the 9 attributes are specified. Of these 683 observations, 239 correspond to *positive* (i.e., malignant breast cancer) cases, while 444 correspond to *negative* (i.e., benign breast cancer) cases. The 9 variables of the problem, clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses, take values from 1 to 10, and will be denoted in this paper by $x_1,\ldots,x_9$.

The *Logical Analysis of Data* (*LAD*), introduced in [12], [14], is a combinatorial optimization-based approach to the analysis of a dataset $\Omega = \Omega^+ \cup \Omega^-$, consisting of "positive" and "negative" observations, each of which is represented as a vector of n attribute values, each attribute x_i taking the values $\{0,1, \ldots, k_i\}$. It has been established in previous studies ([4], [11]) that *LAD* provides a competitive classification tool, comparable in efficiency with the best classification techniques available. The key concept used in *LAD* is that of *patterns*, which makes it in particular possible to provide clear justifications of the reasons for which a not-yet-seen observation should be considered positive or negative. As it will be seen in this paper, the set of patterns can provide numerous other important information about the data. The essence of *LAD* consists in the systematic and exhaustive generation of all the patterns satisfying certain limiting conditions, and the extraction of information from this set of patterns.

Let us first introduce some basic concepts. A set of bounding restrictions imposed on the values of several attributes is called a *conjunction*. A conjunction is called a *positive* (or *negative*) *pattern* if it is sufficiently "biased", i.e., it satisfies at least a prescribed proportion of the positive (negative) observations, and at most a prescribed proportion of the negative (positive) observations. A positive (negative) pattern is called *pure* if every negative (positive) observation in the dataset violates at least one of its defining conditions. For illustration, the conditions "*uniformity of cell size* ≥ 5", "*marginal adhesion* ≥ 2", and "*normal nucleoli* ≥ 3" imposed on the attribute values of the observations in **bcw** represent a pure positive pattern, say P, because they are simultaneous satisfied by 137 (57.3%) of the 239 positive cases, and by none of the negative cases.

It was shown in [11], [15] that the key role in knowledge extraction by *LAD* is played by two types of patterns, called "prime" and "spanned". A *prime pattern* is in fact a "minimal" one, i.e., it is a pattern such that if any of its defining conditions is *relaxed*, the remaining system of conditions defines a conjunction which is no longer a pattern. In the case of a positive (negative) pattern, this means that after the relaxation of any of its defining conditions, the proportion of negative (positive) observations satisfying the remaining conjunction violates the prescribed bias. For illustration, if we consider positive patterns to have a bias of 0% (i.e., no negative observation should satisfied them)

then the pure positive pattern *P* is prime, since by relaxing any of its defining conditions allows some of the negative observations to satisfy its conditions; for instance, by relaxing the second condition to "*marginal adhesion* ≥ 1", and leaving the other two unchanged, the new conjunction *C* obtained in this way will be satisfied by 2 negative observations. A *spanned pattern* is in fact also a "minimal" one, but in a different sense then in the prime case. More exactly, a spanned pattern is such that by *strengthening* any of its defining conditions, at least one of the observations *covered* by the pattern (i.e., satisfying its defining conditions) will no longer be covered. For illustration, if in the positive pure pattern discussed above we strengthen any of its defining conditions, some of the positive observations covered by it will be uncovered; for instance, by strengthening the second condition to "*marginal adhesion* ≥ 3", the number of positive observations covered by the conjunction obtained in this way, say *D*, will drop from 137 to 123. An intuitive example of a prime and a spanned pattern in a fictitious example of a dataset with two variables is shown in Figure 1; the "box" enclosed by the dotted line represents a prime pattern, while that enclosed by a heavy line represents a spanned pattern.

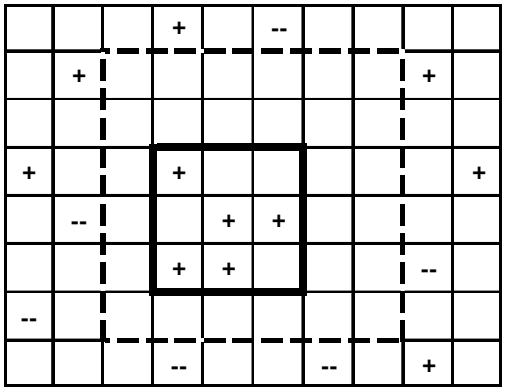

Figure 1

There are five basic parameters s, t, d, p, c associated to a pattern. The *sign* s of a pattern, can only take the values + or – to indicate whether it has a positive or a negative bias. The *type* of the pattern, t, can take the values "prime" or "spanned". The *degree* d of a pattern indicates the number of variables used in its definition, for instance $d = 3$ for the pattern *P* mentioned above. The *prevalence* p of a pattern represents the proportion of observations covered within Ω^+ (in the case of positive patterns), or Ω^- (in the case of negative patterns) covered by it; e.g., the prevalence of *P* is $p = 57.3\%$. In the case of positive (negative) patterns, their *homogeneity* c represents the proportion of those of the observations covered by them which are positive (respectively, negative). For instance the conjunction *C* mentioned before covers 143 positive and 2 negative observations; the corresponding positive pattern has a homogeneity c of 98.62%.

It is important to notice that by definition, the prime patterns are maximal, i.e., no constraint of the type $x \le a$ appearing in their definition can be relaxed to $x \le a'$ with $a < a'$. On the other hand, neither prime nor spanned patterns have to be maximal. If maximality is defined in terms of the set of data points covered. Because of the large number of possible patterns, we usually list only the coveragewise maximal ones. In particular, Tables 5 and 6 only list the coveragewise maximal patterns in the respective pandects.

Both prime and spanned patterns provide valuable information about the dataset. There are advantages and disadvantages associated both to the use of prime and to the use of spanned patterns. Prime patterns of low degree can be generated more efficiently than spanned patterns and are more easily comprehensible. In classification problems in which the primary criterion is to not leave any

new observation "unclassified", they are the preferred tool. On the other hand, spanned patterns usually have higher degrees, and are more "conservative" than the prime ones. It was shown in [7] that classifications based on models using spanned patterns are more "robust", having usually fewer errors, at the cost of leaving somewhat larger numbers of observations unclassified. It will be seen below that prime patterns are the major tool used in the proposed approach to class discovery, while the proposed attribute analysis procedures are mostly based on information provided by spanned patterns.

It is clear that the "most interesting" patterns are those which have *low degrees* (i.e., high explicative power due to their intuitive nature), *high prevalences* (i.e., high reliability, due to the frequency of observations covered by them), and *high homogeneity* (i.e., highly informative, due to their strong bias). The set of all patterns of a given sign s^*, type t^*, degree at most d^*, prevalence at least p^*, and homogeneity at least c^* will be called the $(s^*, t^*, d^*, p^*, c^*)$-*pandect* of the dataset Ω. It should be noted that beside the positive and the negative pandects (for which s^* is "+" or "-", respectively), we shall also consider the pandect of all positive or negative patterns with the given characteristics; we shall use the symbol "$\pm$" to indicate this type of pandect.

It has been shown in [7], [8] that the $(s^*, t^*, d^*, p^*, c^*)$-pandect can be generated (for a fixed d^*) in polynomial time in the input size (i.e., the number n of variables and the number m of observations). If d^* is not fixed, this set can still be generated in total polynomial time (i.e., in n, m and the number c of conjunctions). The generation algorithms of prime and spanned patterns will be briefly outlined in the next section.

In numerous case studies ([1], [2], [5], [6], [9], [11], [13], [20], etc) it was noticed that the most informative pandects were those with d^* equal 2 or 3, or (rarely) 4; usually $d^* \geq 5$ lead to a phenomenon resembling statistical overfitting. The values of p^* had substantial variations (5%-80%) in the case studies examined, being usually in the 10% - 50% range. Finally, c^* was usually restricted to the 90%-100% range, reaching however, in the extreme case of "inseparable" and "unbalanced" data ([6], [20]) values as low as 16.5%.

Since the patterns can be seen as new, synthetic attributes, the pandect defines a new way of representing observations. Each observation can be represented in the *pattern space* corresponding to a pandect as a binary vector, whose j^{th} component indicates whether the observation is or is not covered by the j^{th} pattern. Similarly, the attributes can be represented as ternary vectors in another pattern space corresponding to a pandect, the (-1,0,1) component j of which indicates whether the attribute's values are bounded from above / are unrestricted / are bounded from below in the definition of pattern j.

Pandect-based classification is one of the essential and well-known applications of *LAD* (see e.g., [2], [5], [6], [11], [13], [20]). The purpose of this paper is to describe two other applications of *LAD*, one being a new *class discovery* technique based on clustering in the binary pattern space of a pandect, and the other being an *attribute analysis technique*, based on the role of variables in the corresponding ternary pattern space.

2. Pandect Generation

2.1. Generation of Prime Patterns

The algorithm evaluates the (positive and negative) prevalence, degree, and homogeneity of all possible conjunctions (intervals) in the dataset, and selects subsequently only those which satisfy the pandect-defining degree, prevalence, and homogeneity requirements.

The basic component of the algorithm is the calculation of the positive and of the negative prevalences of each interval in the discrete space. While the algorithm for calculating these

prevalences is given in [8] for the general *n*-dimensional case, its basic principles will be illustrated here for the special case $n = 2$. The attributes are denoted *X* and *Y*, and take values in the sets {0,1,…,*p*} and {0,1,…,*q*}, respectively. We shall denote an *interval* in the two dimensional discrete space as $I = [(i, j),(k,l)]$ ($i \le k$, $j \le l$), and shall indicate by $Cov^{+}(I)$ and $Cov^{-}(I)$ the number of positive, respectively negative, elements of the dataset contained in *I*.

We shall illustrate the way the algorithm calculates prevalences on the positive case. The algorithm starts by associating to Ω the matrix *M*, whose entries are $m_{ij} = Cov^{+}([(i, j),(i, j)])$, and aims at calculating a sequence of matrices leading to a matrix $R^{(0,0)}$ *w*hose elements are the positive prevalences of each of the discrete intervals. In most cases, m_{ij} takes the values 0 or 1, since it is assumed that there is no duplication of observations.

For every pair of integers $k \in \{0,1,..., p\}$ and $l \in \{0,1,..., q\}$, the algorithm recursively constructs matrices $R^{(k,l)}$, where for each $i \le k$, $j \le l$, the entry (*i*,*j*) of $R^{(k,l)}$ is defined as $r_{i,j}^{(k,l)} = Cov^{+}([(i, j),(k,l)])$. The first matrix, $R^{(p,q)} = R$ can be calculated as follows:

R:=*M*;
For *i*:=*p* downto 0 do
For *j*:=*q*-1 donwto 0 do $r_{i,j} := r_{i,j+1} + r_{i,j}$;
For *j*:=*q* donwto 0 do
    For *i*:=*p*-1 downto 0 do $r_{i,j} := r_{i+1,j} + r_{i,j}$;

The recursive computation of $R^{(k,l)}$ is based on the enumeration of the pairs (*k*, *l*), $k \in \{0,1,..., p\}$ and $l \in \{0,1,..., q\}$, by utilizing a generalized form of Gray code ([18]), which makes it possible: (*a*) to generate every pair between (0,0) and (*p*,*q*) exactly once, and (*b*) to assure that any two consecutive pairs differ in exactly one component, and by exactly one unit. As soon as the matrix $R^{(p,q)}$ is computed, the recursion can proceed to the next pair in the Gray sequence. Denoting the current pair by (*k*, *l*), the next element if the Gray sequence is defined as one of pairs (*k*+1, *l*), (*k*-1, *l*), (*k*, *l*+1), (*k*, *l*-1). In the case when the next pair is (*k*, *l*-1), the corresponding matrix is calculated as follows:

R:=$R^{(k,l)}$;
For *i*:=0 to *p* do
{For *j*:=0 to *l*-1 do $r_{i,j} := r_{i,j} - r_{i,l}$;
    For *j*:=*l* to *q* do $r_{i,j} := r_{i,j} + r_{i,l-1}$; }

The calculation of the following matrix in the other three cases is defined in a similar way.

Example. Let $p = 3$, $q = 3$, and let Ω be a dataset. The number m_{ij} of positive elements (*i*, *j*) in Ω is shown in the matrix *M*. The Gray code defines the sequence: (3,3), (3,2), (3,1), (3,0), (2,0), (2,1), … , (0,0). The corresponding matrices are $R^{(3,3)}, R^{(3,2)}$, … $R^{(0,0)}$. We start by calculating $r_{i,j}^{(3,3)} = \sum_{s=i}^{3}\sum_{t=j}^{3} m_{st}$; for example, $Cov^{+}[(1,1),(3,3)] = r_{1,1}^{(3,3)} = 8$. In the following step we calculate the matrix $R^{(3,2)}$, etc.

$$M = \begin{pmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \\ m_{30} & m_{31} & m_{32} & m_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 0 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

$$R^{(3,3)} = \begin{pmatrix} 17 & 11 & 7 & 3 \\ 13 & 8 & 4 & 1 \\ 10 & 7 & 4 & 1 \\ 3 & 2 & 1 & \mathbf{0} \end{pmatrix}, \; R^{(3,2)} = \begin{pmatrix} 14 & 8 & 4 & 7 \\ 12 & 7 & 3 & 4 \\ 9 & 6 & 3 & 4 \\ 3 & 2 & \mathbf{1} & 1 \end{pmatrix}, \ldots, \; R^{(0,0)} = \begin{pmatrix} \mathbf{1} & 1 & 2 & 4 \\ 3 & 4 & 5 & 7 \\ 5 & 8 & 11 & 14 \\ 6 & 10 & 14 & 17 \end{pmatrix}.$$

Making a similar calculation for the negative prevalences of each interval, we have all the information necessary to enumerate all the positive patterns satisfying the prescribed requirements.

The number of operations (additions) *Op* necessary to generate all the prime patterns satisfies $Op \leq 2^n c$, where *c* is the number of conjunctions. Clearly, if the number *n* of attributes is small, the algorithm is linear in the size of the output.

In order to illustrate the efficiency of the algorithm for generation of prime patterns, we present in Table 1 the number of patterns which are generated in the **bcw** dataset for $d^* = \{2, 3, 4\}$, $p^* = 10\%$ and $c^* =$ 90%, 95%, 100%, as well as their generation time on a Pentium III 1GHz processor.

Homogeneity	*Degree*								
	2			*3*			*4*		
	# positive patterns	*# negative patterns*	*Time (s)*	*# positive patterns*	*# negative patterns*	*Time (s)*	*# positive patterns*	*# negative patterns*	*Time (s)*
90%	45	56	0	45	105	1	45	110	43
95%	59	54	0	87	216	1	99	411	43
100%	39	7	0	194	68	1	464	201	43

Table 1. Number and generation time of prime patterns with prevalence ≥ 10% in **bcw** dataset

2.2. Generation of Spanned Patterns

An incremental polynomial time algorithm using a consensus-type approach was developed in [7] for the generation of all the spanned patterns in a dataset.

The well-known method of consensus was proposed by Blake ([10]) and Quine ([21]) for finding prime implicants of a Boolean function.

Malgrange ([19]) used a consensus-type approach to find all the maximal submatrices consisting of the 1's of a 0-1 matrix. Also, a consensus-type algorithm for finding all maximal bicliques of a graph was presented in [3].

Consensus-type methods enumerate the family of all maximal objects of a certain collection, by starting from a sufficiently large set of objects, and systematically completing it by the application of two simple operations. The operation of *consensus adjunction* associates to a pair of objects in the given collection, one or more new objects, and adds them to the collection. The operation of *absorption* removes from the collection those objects which are "dominated" by other objects in the collection. The two operations are repeated as many times as possible, leading eventually to a collection consisting exactly of all the maximal objects.

Below we describe the proposed algorithm ([7]) for the generation of all the positive spanned patterns in a dataset. The negative spanned patterns can be generated in a similar way.

We shall briefly present here a particular variant of the consensus method, which produces the complete collection of all its spanned positive patterns, restricting our attention only to the case of pure patterns. For any two integers a_i, b_i with $0 \le a_i \le b_i \le k_i$, we denote by $[a_i, b_i]$ the interval of integers $\{a_i, a_i+1, \ldots, b_i\}$, and we represent an n-dimensional interval as $[a_1, b_1] \times \ldots \times [a_n, b_n]$. Let $P=[a_1, b_1] \times \ldots \times [a_n, b_n]$ and $P'=[a'_1, b'_1] \times \ldots \times [a'_n, b'_n]$ be a pair of positive pure spanned patterns, and let P'' be the interval $[a''_1, b''_1] \times \ldots \times [a''_n, b''_n]$, where $a''_i = \min\{a_i, a'_i\}$ and $b''_i = \max\{b_i, b'_i\}$, $i = 1, \ldots, n$. If P'' is a positive pure pattern, then it is called the *consensus* of the pure patterns P and P'. Clearly, a pair of positive spanned pure patterns can have at most one consensus, which is the pure pattern spanned by the observations covered by P or P'. We say that the positive spanned pure pattern P *absorbs* the positive spanned pure pattern P' if $P = P'$.

The proposed algorithm for generating all positive spanned pure patterns runs as follows. Let us start with the collection $C := C_0$ of all those spanned patterns, which cover exactly one observation in Ω^+. The collection C will be augmented by the inclusion of additional patterns. More precisely, at each stage, we shall select a pair of patterns P_0 in C_0 and P in C, whose consensus P' exists and is not already contained in C, and add P' to C. This operation will be repeated until the collection C cannot be enlarged anymore in this way. It was proved in [7] that at this stage, C will consists exactly of the collection of all spanned patterns.

Example. We shall illustrate the algorithm for the dataset Ω consisting of the four observations $w_1 = (1,0,2)$, $w_2 = (0,2,0)$, $w_3 = (3,1,1)$, $w_4 = (2,0,2)$, where w_1, w_3, and w_4 are positive, and w_2 is negative. The input collection C_0 is $\{P_1 = [1,1] \times [0,0] \times [2,2], P_3 = [3,3] \times [1,1] \times [1,1], P_4 = [2,2] \times [0,0] \times [2,2]\}$. Initialize $C := C_0$. Perform consensus adjunction for the pair of patterns P_1 in C_0 and P_3 in C: the candidate for consensus is $P_{1,3} = [1,3] \times [0,1] \times [1,2]$, covering the positive observations ω_1, ω_3, ω_4; since $P_{1,3}$ is not contained in C, it is added to C. Perform consensus adjunction for the pair of patterns P_1 in C_0 and P_4 in C: the candidate for consensus is $P_{1,4} = [1,2] \times [0,0] \times [2,2]$, covering the positive observations ω_1, ω_4 ; since $P_{1,4}$ is not contained in C, it is added to C. Perform consensus adjunction for the pair of patterns P_3 in C_0 and P_4 in C: the candidate for consensus is $P_{3,4} = [2,3] \times [0,1] \times [1,2]$, covering the positive observations ω_3, ω_4 ; since $P_{3,4}$ is not contained in C, it is added to C. The consensus of any other pair of patterns from C and C_0 is contained in C. The algorithm stops and outputs the family of all positive pure spanned patterns $C = \{P_1, P_3, P_4, P_{1,3}, P_{1,4}, P_{3,4}\}$.

The proof of correctness of this algorithm (i.e. of the fact that it stops after a finite number of steps, coinciding at termination with the list of all pure spanned positive patterns), as well as its worst-case complexity, are presented in [7]. The algorithm runs in incremental polynomial time, its total running time being $O(bm_+(m+m_+n))$, where b is the number of positive spanned pure patterns, and m_+ is the number of positive observations in the dataset. In order to illustrate the efficiency of the proposed algorithm, we present in Table 2 the results of applying it to the **bcw** dataset.

Prevalence	*Number of Spanned Pure Patterns Produced*			
	1000	5000	10000	16000
5%	3	25	47	93
10%	4	28	51	99

Table 2. Generation time (in seconds) of spanned pure patterns in **bcw** dataset

In all real-life applications encountered, the number of spanned pure patterns was extremely high. In view of this fact, it was important to apply various filtering mechanisms to restrict the number of pure patterns produced, and to keep in this way both time and memory requirements at an acceptable level. The final list of pure spanned patterns is obtained from the list *C*, based on several selection criteria, which include restrictions on the number of pure patterns produced, and the total time allocation.

3 Applications of patterns

One of the most important applications of the collections of positive and negative patterns is the construction of classifiers.

3.1. Pattern-Based Clustering

In their original form, the observations in the dataset $\Omega = \Omega^+ \cup \Omega^-$ appear as numerical and binary vectors in the *n*-dimensional discrete space. The similarities and dissimilarities existing between the observations represented in this way are frequently explored ([16], [22]) by using various clustering techniques, by partitioning Ω into subsets, or "clusters", in such a way that the pairs of observations grouped into a common cluster have a high "degree of similarity" (according to various metrics), and the pairs of observations belonging to different clusters have high "degrees of dissimilarity".

The representation of observations in pattern space, discussed at the end of Section 1, makes it possible to use clustering techniques for the exploration of similarities between observations which are covered by similar sets of patterns. Given the pandect $\Pi = \{P_1, P_2, ..., P_r, N_1, N_2, ..., N_s\}$, each observation in $w \in \Omega$ is represented as an (*r*+*s*)-dimensional binary vector, which indicates the patterns in Π which cover w. The interest in clustering Ω in the space of patterns comes from the fact that patterns can be viewed as synthetic attributes which reflect more closely the positive or negative nature of an observation than the original attributes. Therefore, it can be expected that observations covered by the same (or almost the same) sets of patterns may have high degrees of similarity.

In order to compare the usefulness of clustering in pattern space vs. clustering in the original discrete space, we have carried out several *k*-means clustering experiments (for k = 2, 3 and 4) using the two representations of the observations. In each experiment we measured the percentage of those pairs of observations *x*, *y*, (with $x \in \Omega^+$, $y \in \Omega^-$) which were assigned to different clusters, and found the following results:

	k-Means Clustering		
	k = 2	*k = 3*	*k = 4*
Attribute Space	91.15%	95.28%	95.07%
Pattern Space	95.95%	96.54%	97.29%

Table 3. Dissimilarities for *k*-Means Clustering

It is clear that clustering in pattern space produces consistently a better separation of positive and negative points than clustering in the original discrete space of the attributes.

Based on this conclusion, we have carried out a series of pattern space-based clustering experiments on several publicly available datasets. We shall illustrate the type of results found in

such applications on the **bcw** dataset, using the (+, *p*, 3, 30%, 100%) - and the (-, *s*, 3, 50%, 100%) - pandects (see Tables 5, 6). Because of space limitation, we shall only present here conclusions concerning the set Ω^+ of positive observations, although similar conclusions were also found for Ω^-.

(a) Stability Applying 3-means clustering to Ω, we find that one of the three clusters, say *C*, contains 105 positive observations and no negative ones, while the remaining observations are clustered into a set *D* consisting of 404 negative observations, and no positive ones, and a set *E* containing both positive and negative observations.

The repeated application of *k*-means clustering produces usually partitionings which differ from one experiment to the other. This is a consequence of the way k-means clustering works. It starts with a random partition of the original set into *k* subsets, and is followed by a sequence of transfers of elements between the subsets. In view of this fact, it is extremely surprising that the set *C* turns out to be very "stable". Indeed, by repeating the same clustering 20 times, we find that the cluster *C* reappears without any changes in every single experiment.

Even more unexpected is the fact that a 2-means clustering of Ω^+ splits the set of positive observations into two clusters, one of which, say C^*, is almost exactly the set *C*, differing from it only by the addition of one single supplementary observation. Again, 20 repetitions of this experiment, produce invariably C^* as one of the two clusters.

Applying the more robust *partitioning k-medoids* clustering technique ([17]), the same property reappears: in each of 20 experiments, the set C^* turns out to be one of the clusters appearing among the final clusters.

Finally, applying now a third clustering technique, called *hierarchical agglomerative* clustering ([17]), in which the number of subsets in the partition is not fixed *a priori*, the set C^* reappears again in all the 20 repetitions of the experiment.

(b) Strong positivity Within the set of positive observations Ω^+, the subset C^* displays a series of powerful characteristics indicating the positive nature of its observations, distinguishing it clearly from $C' = \Omega^+ \setminus C^*$.

First, the proportion of positive observations in the neighborhood of each observation in C^* is much higher than in the neighborhood of observations in *C*'. In order to see this, the original data were "normalized", i.e., the measurement x_{ij} of each attribute *j* in observation *i* was replaced by $(x_{ij}-m_j)/s_j$, where m_j and s_j are respectively the mean and the standard deviation of attribute *j* in Ω. Using the Euclidean metric, the spheres of radius *i* centered in the points of C^* and of *C*' contain on the average the following proportions of positive observations:

R	*Average proportion of + in C' (%)*	*Average proportion of + in C'' (%)*
1.50	99	35
1.75	98	35
2.00	98	37
2.50	98	45
3.00	96	54
3.50	95	52
4.00	90	46

Table 4. Positive content of nearest neighborhood

Clearly, the proportion of positive points in the neighborhoods of the points in C^* exceeds substantially that of the points in the neighborhoods of points in C', regardless to the choice of the neighborhood defining radius. Moreover, the disproportion increases rapidly when the radius decreases.

Second, the points in C^* have a much stronger coverage by positive patterns than those in C'. Indeed, the proportion of positive patterns in Π covering an average point in C^* is 3 times higher (57%) than in C' (19%). Moreover, the average prevalence of the patterns covering the points in C^* is significantly higher (40%) than in C' (32%).

(c) Separation The observations in C^* can be separated from the other observations in Ω by the interval hull $[C^*]$ of C^*, i.e., the unique minimum n-dimensional interval which includes C^* in the original discrete space. Indeed, the set $[C^*]$ consists of C^*, 9 addition points in Ω^+, and 1 point in Ω^-.

In fact, $[C^*]$ can be viewed as a spanned pattern of high prevalence (48.12%), and homogeneity (99.14%). Using the variables $x_1,\ldots, x_9$ defined in Section 1, this pattern can be described by the system of inequalities "$x_3 \geq 2$, $x_4 \geq 6$, $x_5 \geq 2$, and $x_7 \geq 2$". It should be remarked that after the elimination of redundancies, the resulting prime pattern "$x_3 \geq 2$, $x_4 \geq 6$" covers the same points as $[C^*]$. Finally, the Fisher linear discriminant

$$38.7\, x_1 - 9.6\, x_2 + 29.3\, x_3 - 49\, x_4 + 5.4\, x_5 - 22.4\, x_6 + 14\, x_7 - 9.7\, x_8 + 1.9\, x_9 + 2334$$

separates the entire interval $[C^*]$ from $\Omega \setminus C^*$ with an accuracy of 99.4%.

(d) High predictability The most important consequence of the stability, the strong positivity, and the separability of $[C^*]$ is that the classification by *LAD* of the positive points belonging to this set is extremely accurate. Indeed, the application of 20 cross-validation experiments by 2-folding, produced 4.35% errors in the set $\Omega^+ \setminus [C^*]$, but only 0.67% errors in $[C^*]$.

Pattern		*Pattern Description*									*Prevalence (%)*	
Name	*Sign*	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	*Positive*	*Negative*
P1	+		>4					>4			54	0
P2	+			>4				>4			52	0
P3	+		>4		>2				>2		51	0
P4	+			>2	>5.5						46	0
P5	+		>3		>5.5						46	0
P6	+	>6.5	>4								44	0
P7	+		>4			>3			>4.5		44	0
P8	+	>3.5	>2		>5.5						42	0
P9	+	>3.5			>5.5			>2			42	0
P10	+				>5.5	>3		>2			41	0
P11	+			>5.5	>2				>3.5		40	0
P12	+		>2		>5.5	>3					40	0
P13	+	>6.5		>4			>2.5				40	0
P14	+				>5.5		>4.5	>2			39	0
P15	+		>2		>5.5		>4.5				39	0
P16	+			>4			>4.5		>4.5		38	0
P17	+			>4	>4				>2		38	0
P18	+		>4				>4.5		>4.5		38	0
P19	+				>5.5			>4			38	0
P20	+			>5.5		>3			>4.5		37	0
P21	+				>5.5		>5.5				37	0
P22	+		>4			>5					36	0
P23	+	>3.5			>5.5				>2		36	0
P24	+				>5.5	>3			>2		36	0
P25	+		>4							>1.5	36	0
P26	+				>5.5		>4.5		>2		34	0
P27	+			>4						>1.5	33	0
P28	+	>6.5		>4		>3					33	0
P29	+	>6.5		>4					>3.5		31	0
P30	+		>3							>2	30	0
N1	-		≤3				≤2.5		≤3.5		0	91
N2	-		≤3			≤3.5	≤2.5				0	90
N3	-		≤3	≤3			≤2.5				0	90
N4	-	≤6.5				≤3.5	≤2.5				0	89
N5	-			≤3			≤2.5			≤4	0	89
N6	-	≤6.5	≤2				≤4.5				0	89
N7	-		≤4				≤2.5		≤2		0	89
N8	-			≤4			≤2.5		≤2		0	88
N9	-	≤5	≤3				≤2.5				0	88
N10	-	≤5		≤3			≤2.5				0	86
N11	-		≤2				≤2.5			≤2.5	0	86
N12	-		≤2				≤2.5	≤4			0	85
N13	-		≤4			≤2.5	≤2.5				0	85
N14	-	≤5		≤2			≤4.5				0	85
N15	-			≤4		≤2.5	≤2.5				0	85
N16	-		≤2			≤2.5	≤4.5				0	84
N17	-					≤2.5	≤2.5	≤4			0	84
N18	-		≤2	≤2			≤4.5				0	83
N19	-	≤6.5			≤2.5	≤2.5					0	83
N20	-			≤2		≤2.5	≤4.5				0	80
N21	-	≤5		≤2	≤2.5						0	80
N22	-		≤4				≤4.5	≤2.5			0	66
N23	-			≤5.5			≤4.5	≤2.5			0	66
N24	-	≤6.5					≤4.5	≤2.5			0	66
N25	-		≤5.5				≤2.5	≤2.5			0	64
N26	-						≤2.5	≤2.5	≤5.5		0	64
N27	-					≤5	≤2.5	≤2.5			0	63
N28	-	≤6.5			≤2.5			≤2.5			0	61
N29	-	≤3.5	≤3	≤3							0	61
N30	-	≤3.5		≤3			≤5.5				0	61
N31	-	≤3.5	≤3				≤4.5				0	60
N32	-	≤3.5			≤5.5		≤4.5				0	60
N33	-	≤3.5		≤3		≤3.5					0	60
N34	-	≤3.5					≤5.5		≤2		0	59
N35	-	≤3.5		≤3				≤4			0	59
N36	-	≤3.5	≤2								0	59
N37	-	≤3.5		≤4	≤2.5						0	58

Table 5. Pandect of pure prime patterns having degrees ≤ 3

and positive (negative) prevalences ≥ 30% (respectively, 50%)

Pattern		*Pattern Description*									*Prevalence (%)*	
Name	*Sign*	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	*Positive*	*Negative*
P1	+		>4					>4			54	0
P2	+		>2	>4				>4			52	0
P3	+		>4		>2.5			>2.5	>2		51	0
P4	+		>2	>2	>5.5						46	0
P5	+		>3		>5.5						46	0
P6	+	>6.5	>4								44	0
P7	+		>4			>2.5			>4.5		44	0
P8	+	>2.5			>5.5			>2.5			42	0
P9	+	>2.5		>4	>2.5				>2.5		42	0
P10	+		>2	>4	>2.5		>4.5		>2		42	0
P11	+	>2.5	>2		>5.5						42	0
P12	+				>5.5	>2.5		>2.5			41	0
P13	+		>2	>5.5	>2.5	>2.5		>2.5	>2.5		40	0
P14	+	>6.5		>4			>2.5				40	0
P15	+		>2		>5.5	>2.5		>2.5			40	0
P16	+		>2		>5.5		>4.5				39	0
P17	+				>5.5		>4.5	>2.5			39	0
P18	+		>2	>5.5	>2.5	>2.5		>2.5	>2		39	0
P19	+		>2	>5.5	>2.5	>2.5	>2.5	>2.5	>2		39	0
P20	+		>4	>2		>2.5	>4.5		>4.5		38	0
P21	+		>2	>4	>4			>2.5	>2		38	0
P22	+		>2	>4			>4.5		>4.5		38	0
P23	+				>5.5			>4			38	0
P24	+		>2	>5.5		>2.5			>4.5		37	0
P25	+				>5.5		>5.5				37	0
P26	+		>4			>5					36	0
P27	+	>2.5			>5.5				>2		36	0
P28	+				>5.5	>2.5		>2.5	>2		36	0
P29	+		>4							>1.5	36	0
P30	+	>5	>4			>2.5			>2		35	0
N1	-		≤3				≤2.5		≤3.5		0	91
N2	-	≤6.5	≤3				≤2.5		≤3.5		0	90
N3	-		≤3			≤3	≤2.5				0	90
N4	-		≤3				≤2.5		≤3.5	≤2	0	90
N5	-		≤4			≤3	≤2.5			≤4	0	90
N6	-		≤3	≤3			≤2.5				0	90
N7	-	≤6.5			≤4		≤2.5		≤3.5		0	89
N8	-	≤6.5				≤3	≤2.5				0	89
N9	-	≤6.5	≤2				≤4.5				0	89
N10	-		≤3	≤3			≤2.5			≤4	0	89
N11	-		≤3				≤2.5	≤4	≤3.5		0	89
N12	-		≤3	≤4		≤3	≤2.5			≤4	0	89
N13	-	≤5	≤4				≤4.5		≤2		0	89
N14	-	≤6.5	≤3	≤3			≤2.5				0	89
N15	-	≤5		≤4			≤4.5		≤2		0	89
N16	-	≤6.5		≤3	≤4		≤2.5				0	89
N17	-		≤3				≤2.5		≤2		0	89
N18	-	≤6.5		≤3		≤5	≤2.5				0	89
N19	-	≤5			≤4		≤4.5		≤2		0	89
N20	-		≤3	≤4			≤2.5		≤2		0	88
N21	-		≤4			≤3	≤2.5	≤4		≤4	0	88
N22	-	≤6.5		≤3			≤2.5		≤3.5		0	88
N23	-	≤5	≤3				≤2.5				0	88
N24	-		≤2	≤3		≤5	≤4.5				0	88
N25	-	≤5	≤4		≤4		≤2.5				0	88
N26	-	≤6.5		≤3			≤2.5	≤5			0	88
N27	-	≤6.5	≤3				≤2.5		≤2		0	88
N28	-		≤2	≤3			≤4.5	≤5			0	88
N29	-		≤2	≤4		≤5	≤4.5		≤2		0	87
N30	-		≤3				≤2.5	≤4	≤2		0	87
N31	-	≤5				≤3	≤4.5		≤2		0	87
N32	-		≤4		≤4	≤2.5	≤4.5		≤3.5		0	87
N33	-	≤5			≤4	≤5	≤2.5				0	87
N34	-		≤4	≤4	≤4	≤2.5	≤4.5		≤3.5		0	87
N35	-	≤5			≤4		≤2.5			≤2	0	87
N36	-		≤3			≤5	≤2.5	≤4	≤2		0	87
N37	-		≤2	≤4			≤4.5	≤4	≤2		0	87

Table 6. Pandect of pure spanned patterns

having positive (negative) prevalences ≥ 35% (respectively, 85%)

3.2. Attribute Analysis

An interesting application of the availability of the pandect is the possibility it offers for measuring the relative importance of the various attributes, as well as for identifying monotone attributes. We shall illustrate this possibilities on the 9 attributes of the **bcw** dataset.

3.2.1. Importance of Attributes

The degree of participation of a variable in the patterns appearing in the pandect offers a valuable measure of its importance. Clearly, this analysis has to be based on spanned, rather than prime patterns, since the description of a prime pattern "hides" the implicit bounding conditions on some of the variables.

Assuming that the set of spanned patterns in the pandect $\Pi^* = \{P_1^*, P_2^*, ..., P_u^*, N_1^*, N_2^*, ..., N_v^*\}$, we shall define the role r_j of variable x_j as $w_j/(u+v)$, where w_j is the number of patterns in Π^* which include a bounding condition on x_j. The values of r_j for the 9 variables $x_1, x_2, ..., x_9$ in **bcw** are 65%, 43%, 43.5%, 33.5%, 31.5%, 52%, 43.5%, 35.5% and 23.5%, showing that their respective *ranks* are 1, 5, 3, 8, 6, 2, 4, 7 and 9.

An indication of the significance of this frequency-based ranking of the variables, is shown in the following experiment. We have removed from the dataset the top ranked *k* variables, constructed a *LAD* model on the remaining variables and evaluated through cross-validation the accuracy of the classification provided by the model. After this, the experience was repeated by removing this time the bottom ranked *k* variables. The resulting accuracies are shown in Table 7. The results clearly demonstrate the consistently superior performance of the models built on the high ranking variables compared to those built on the low ranking ones.

k	*Accuracy of Model Built on*	
	Top Ranked k Attributes	*Bottom Ranked k Attributes*
3	81.2	62.9
4	92.5	86.3
5	93.9	94.0
6	95.3	93.2
7	95.7	94.0
8	95.5	93.4
9	95.5	95.5

Table 7. Accuracies of models using high/low ranked attributes

3.2.2. Monotonicity of Attributes

A variable x_j with the properties that: (*i*) no positive pattern in the pandect includes a bounding condition of the form $x_j \leq a$, and (*ii*) no negative pattern in the pandect includes a bounding condition of the form $x_j \geq b$, is called a *positive variable*. Similarly, a variable x_j with the properties that: (*i*) no negative pattern in the pandect includes a bounding condition of the form $x_j \leq a$, and (*ii*) no positive pattern in the pandect includes a bounding condition of the form $x_j \geq b$,

is called a *negative variable*. Positive and negative variables are called *monotone*. Clearly, if x_i is a positive variable, $w = (x_1^*, x_2^*, ..., x_{i-1}^*, x_i^*, x_{i+1}^*, ..., x_n^*) \in \Omega^+$ and if $x_i^{**} \geq x_i^*$, then

$$w' = (x_1^*, x_2^*, ..., x_{i-1}^*, x_i^{**}, x_{i+1}^*, ..., x_n^*) \in \Omega^+ .$$

A similar observation holds, of course, for negative variables. For illustration, we mention that in the **bcw** dataset, all the 9 variables are positive.

References

[1] S. Abramson, G. Alexe, P.L. Hammer, J. Kohn, Using Logical Analysis of Data (LAD) Based Computer Model Predicts Cell Metabolic Activity on Polymeric Substrates, *RUTCOR Research Repor*t, RRR 40-2002; Communication at the 29th Annual Meeting of the Society for Biomaterials, Reno, Nevada, April-May 2003.

[2] G. Alexe, S. Alexe, E. Boros, D. Axelrod, P. L. Hammer. Combinatorial Analysis of Breast Cancer Data from Image Cytometry and Gene Expression Microarrays. *RUTCOR Technical Report*, RTR 3-2002

[3] G. Alexe, S. Alexe, Y. Crama, S. Foldes, P. L. Hammer, B. Simeone. Consensus algorithms for the generation of all maximal bicliques. Rutgers University, *RUTCOR Research Report* RRR 41-2001; *DIMACS Technical Report* 52-2002; *Discrete Applied Mathematics* (in print).

[4] G. Alexe, S. Alexe, P. L. Hammer, A. Kogan. Comprehensive vs. Comprehensible Classifiers in Logical Analysis of Data. Rutgers University, *RUTCOR Research Report*, RRR 9-2002; *DIMACS Technical Report* 2002-49; *Annals of Operations Research* (in print).

[5] G. Alexe, S. Alexe, P. L. Hammer, L. Liotta, E. Petricoin, M. Reiss. Logical Analysis of the Proteomic Ovarian Cancer Dataset. *RUTCOR Technical Report*, RTR 2-2002 (http://rutcor.rutgers.edu/~rrr/rtr/2-2002.pdf).

[6] S. Alexe, E. Blackstone, P.L. Hammer, H. Ishwaran, M.S. Lauer, C.E.P. Snader, Coronary Risk Prediction by Logical Analysis of Data, *Annals of Operations Research*, 119 (2003), pp. 15-42.

[7] G. Alexe, P. L. Hammer. Spanned Patterns in Logical Analysis of Data, *RUTCOR Research Report*, RRR 15-2002, *Annals of Operations Research* (2003), (in print).

[8] S. Alexe, P. L. Hammer, Accelerated Algorithm for Pattern Detection in Logical Analysis of Data. *RUTCOR Research Report*, RRR 59-2001, *Annals of Operations Research* (2003), (in print).

[9] S. Alexe, P.L. Hammer, A. Kogan, M.A. Lejeune, A Non-Recursive Regression Model For Country Risk Rating, *RUTCOR Research Report*, RRR 9-2003.

[10] A. Blake, *Canonical Expressions in Boolean Algebra.* Ph.D. Thesis, University of Chicago (1937).

[11] E. Boros, Hammer, P.L., Ibaraki, T., Kogan, A., Mayoraz, E., Muchnik, I. An Implementation of Logical Analysis of Data. *IEEE Transactions on Knowledge and Data Engineering*, 12 (2) (2000), 292-306.

[12] Y. Crama, P.L. Hammer, T. Ibaraki, Cause-Effect Relationships and Partially Defined Boolean Functions. *Annals of Operations Research* 16 (1988), 299-326.

[13] A. Hammer, P.L. Hammer, I. Muchnik. Logical Analysis of Chinese Productivity Patterns, *Annals of Operations Research*, 87 (1999), pp. 165-176.

[14] P.L. Hammer, Partially defined Boolean functions and cause-effect relationships, *International Conference on Multi-Attribute Decision Making Via OR-Based Expert Systems*, University of Passau, Passau, Germany, (1986).

[15] P. L. Hammer, A.Kogan, B. Simeone, and S. Szedmak, Pareto-Optimal Patterns in Logical Analysis of Data, *RUTCOR Research Report*, RRR 7-2001, *Discrete Applied Mathematics*, (fouthcoming).

[16] J.A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, Inc. (1975).
[17] L. Kaufman, P.J. Rousseeuw, *Finding Groups in Data: An introduction to Cluster Analysis*, John Wiley & Sons, Inc. (1990)
[18] Y. Koda, F. Ruskey, A Gray code for the ideals of a forest poset. *Journal of Algorithms* 15 (1993), pp. 324-340.
[19] Y. Malgrange, *Recherche des sous-matrices premières d'une matrice à coefficients binaires. Applications à certains problèmes de graphe.* Deuxième Congrès de l'AFCALTI, October 1961, Gauthier-Villars, (1962), pp. 231-242.
[20] M.S. Lauer, S. Alexe, C.E.P. Snader, E. Blackstone, H. Ishwaran, P. L. Hammer. Use of the "Logical Analysis of Data" Method for Assessing Long-Term Mortality Risk After Exercise Electrocardiography. *Circulation*, 106 (2002), pp. 685-690.
[21] W. Quine, A way to simplify truth functions. *American Mathematical Monthly*, 62, (1955), 627-631.
[22] A. Struyf, M. Hubert, P.J. Rousseeuw, Integrating Robust Clusteriung Techniques in S-PLUS. Computational Statistics and Data Analysis, 26 (1997), pp. 17-37.