No 734

TALLINNA TEHNIKAÜLIKOOLI TOIMETISED
TRANSACTIONS OF TALLINN TECHNICAL UNIVERSITY

UDC 681.3.06

Rein Kuusik

THE SUPER-FAST ALGORITHM OF HIERARCHICAL CLUSTERING AND THE THEORY OF MONOTONIC SYSTEMS

Abstract

A new hierarchical clustering algorithm based on theory of monotonic systems is described. Let us have $N \cdot M$ data table (N is number of objects, M is number of attributes), where each attribute j, $j = 1,...,M$, has a discrete value $h_j = 0,1,...,K-1$. If $N = K^M$, then the complexity of described algorithm is $O(N^2)$ for extracting all $(K+1)^M$ clusters.

1. The Problem

Most readers are evidently familiar with the following presentation of a problem.

Suppose that the finite discrete data matrix $X(N,M)$ (the set of objects X) of object-attribute type is given, where N is the number of objects (examples) and M is the number of attributes. Every attribute j can acquire integer values in the interval $h_j = 0,1,2,...,K_j - 1$.

Problem: We should find all existing value combinations (VC) of attributes in set X.

For instance, if in $X(N,M)$ for each attribute j $(j = 1,2,...,M)$, $K_j = 2$ and $N = 2^M$, then it is easy to check that the number of really existing VCs on such a set X is equal to 3^M.

Every VC describes a certain subset X_{VC} of objects and subset X_{VC} of elements in set X. The last set is called a cluster in the theory of classification, that is why we call the process of extracting VCs clustering.

A serious problem is encountered when solving analogous problems in the theory of Artificial Intelligence (AI), in the Graphs Theory (GT), Boolean Algebra (BA), etc. For instance, generation of facts and rules from discrete database (AI), finding of cliques (GT), finding of the shortest and minimal disjunctive normal form (BA). Those examples represent only a small part of the bulk of problems in question, most of which are NP-complicated [1]. Therefore an efficient method of finding a solution to the problems attached to it is so important.

In this paper we estimate and prove an algorithm with the complexity of $O(N^2)$ operations to solve the problem described above for finite discrete data matrix $X(N,M)$ with $N = K^M$, $K \geq 2$.

2. **How Has the Problem Been Solved Until Now**?

To solve the problem, two main algorithms are used. Let us denote them B1 and B2, respectively. Before describing them let us define the concept "intersections of sets".

The term "intersection", as we are using it here, corresponds to the classical definition of the Set Theory:

Definition 1.1. Intersection of sets $X1$ and $X2$ is the set of elements $X_{ij} \in X$, which belongs to $X1$ and $X2$, simultaneously.

For instance, if we have a set $X(2,3)$ with objects

1 2 2
1 1 2 ,

Then taking the first object of the set X as a set $\mathrm{X1}$ and the second object of the set X as the set $\mathrm{X2}$, we acquire the set $1*2$ as their intersection $X1 \cap X2$, where the symbol '*' denotes an empty element.

When there are no common elements to be found, an empty set (empty intersection) will be the result of the intersection:

$$\mathrm{X1} \cap \mathrm{X2} = \varnothing .$$

Analogously, we can define intersection over sets $\mathrm{X_i} \subseteq \mathrm{X}$, $\mathrm{i} > 2$ (see definition 3.1).

Now we are ready to describe algorithms B1 and B2 to find real VCs:

For algorithm B1 all real VCs are found interactively as intersections over the object pairs, triplets, quadruples, etc., over N objects;

For algorithm B2 all real VCs are found as intersections over all object pairs, next intersections over all intersections pairs, then intersections over the acquired intersection pairs, etc. until all the intersections found are empty ones.

The schema of those algorithms is very compact (cycles in cycles), but unfortunately this compactness is deceptive. In reality, both algorithms are very hard to use for two reasons:

1) Interceptions of objects happen quite spontaneously, i.e., we cannot say for sure which objects are to be interested to obtain a VC. Because of that a lot of work is done for nothing, since a majority of intersections are either empty or repeating,

2) It is very time-consuming to determine a VC originality, i.e. whether or not it is already generated.

In [2] an algorithm is presented which is considerably easier to work with than the one described above. Let us denote it by B3. If we describe B3 in Pseudo-basic, it will be written as follows:

```
B3: FOR I1 = 1 TO N DO
        FOR I2 = I1 + 1 TO N DO
            {intersection}
            IF intersection = ∅, then NEXT I2
            FOR I3 = I2 + 1 TO N DO
                {intersection}
                IF intersection = ∅, then NEXT I3
                …
                FOR IN = I((N – 1) + 1 TO N DO
                    …
                NEXT IN
            …
        NEXT I2
NEXT I1.
```

As is shown, in each FOR-cycle the next object of data matrix intersects the intersection of the previous FOR-cycle. If the result is an empty intersection, then the given branch is considered to be a dead-end (not all of the following FOR inner cycles are needed, since their result will be always an empty intersection). The existence of such a control allows to practically minimizing the number of empty intersections. This makes algorithm B3 so efficient compared to the first two.

Algorithm B3 has a complexity of $O(A \cdot M)$ operations, where A is the number of non-empty intersections in $X(N, M)$ [2].

Algorithm B3 has one disadvantage – repetition of generated VCs. If we represent the VC generation process as a hierarchical tree then in the case of B3 the finding of a repeating VC is equivalent to the repeated coming to an already passed branch. Here, too, we are unable to say whether subbranches of the branch are already passed or yet to be passed. In the case of B3 to create such a control check is very costly because VC is generated out spontaneously (the objects are intersected in accordance with the sequence in the initial data matrix X allocation).

Algorithm B3 is very time consuming because $A >> L$, where L is the number of real VCs on set X.

In conclusion, the third disadvantage of all the algorithms described should be underlined here.

3) Because of the spontaneous intersection, all of those algorithms are very difficult (if not impossible) to use efficiently for optimizing tasks, where extraction of VC has to be an oriented process.

As seen from the discussions the same problems are encountered that have been described above. To escape those disadvantages we have set up the task of VC extraction differently – to find such an objective function $F(x)$ that permits to determine all really existing VCs on the set $X(N, M)$.

This paper aims to solve this problem.

3. Main Concepts

Having completed the work, we can say that we did not expect the solution to be so difficult to acquire. When eventually a solution was reached, it was clear that the tools necessary for it were discovered from the theory of monotonic systems. Before explaining the theory, we have to specify how the task was set up, and give the definitions and theorems.

3.1. Denotations

Let us denote by

1. X – a set $X = \{X_i\},\ i = 1,2,...,N$, where each object X_i is a conjunction of M attribute values: $X_i = \&_{j=1}^{M} h_j$.

2. H – a value combination (VC) of certain attributes $H = \&_{q \in D} h_q$, $D = \{j_e\}$, $e = 1,...,E_H$ (number of elements h_q in H), $1 \le E_H \le M$, $1 \le j_e \le M$, $j_f, j_t \in D$, $j_f \# j_t, H \subseteq X_i$.

3. Each value combination H defines on set X a subset of objects $X_H = \{X_p\},\ p = 1,2,...,N_H$, $1 \le N_H \le N$, $\{X_p\}$ are all objects $X_i \in X$ that contain H: $X_H = \{X_i \in X \mid X_i \supseteq H\}$.

4. Each value combination H defines on the set X a subset of elements $X^H \subseteq X : X^H = \{X_{ij} \in H, i = 1,2,...,N, j = 1,2,...,M\}$.

3.2. Definitions and Theorems

Definition 3.1. Intersection over a set $Y = \{Y_t\},\ t = 1,2,...,T$, $Y_t = \&h_j$, is a set of such elements h_q which belong simultaneously to all $Y_t : \cap Y = \bigcap_{t=1}^{T} Y_t = \&_q h_q = H$.

In Y for H there exists always a corresponding subset of objects $Y_H = \{ y_p \}, p = 1,..., N_H,\ N_H \le N$.

If $N = 1$, then intersection over Y is an object itself.

If there exist no objects $Y_t \in Y$ for which $H \subset Y_t$, then $Y_H = \varnothing$.

Suppose that an arbitrary VC H is given on X. It describes a subset of objects $X_H \subseteq X$. Then an intersection over X_H equals $A \supseteq H$:

$$X \cap X_H = \bigcap_{p=1}^{N_H} X_p = A,\ A \supseteq H,\ 1 \le N_H \le N.$$

It means that X_H can simultaneously correspond to several VCs $H \subset H_1 \subset H_2 \subset ... \subset A$, i.e. set X_H is defined as not unique. To overcome this situation let us give the following definitions.

Definition 3.2. Elementary conjunction (EC) on X_H is such an intersection over the set X_H, where $\cap X_H = A(\supseteq H),\ X_A = X_H,\ N_H \le N$.

It follows from here that in the case of $A \supset H$, $X_A = X_H$, A not H is an EC.

Definition 3.3. Maximal EC on X_H is such an intersection over X_H in case of which for a VC $H = \&_q h_q$ is a valid relation $\cap X_H = A\left(= \&_e h_e \right) \supset H$, $1 \le q < e \le M,\ X_A = X_H$.

By definition, H is EC if $\cap X_H = H$. H is a maximal EC if it is EC and contains at least on VC $H_t \subset H$ such, that $|X_{H_t}| = |X_H|$ on X. That means that the set of objects $X_H \subseteq X$ is defined unique.

As you can see from definitions an EC H corresponds to the intersection of objects in all algorithms B1 to B3.

Each EC H defines a subset

$$X^H \subseteq X : X^H = \left\{ X_{ij} \in X \,\middle|\, X_{ij} \in H,\ i = 1,2,..., N,\ j = 1,2,..., M \right\}.$$

To show identity of the subset X^H with a kernel of theory of monotonic systems, let us prove some theorems.

Theorem 3.1. To be an elementary conjunction it is sufficient and necessary for a value combination H that there exist at least two objects $X_a, X_b \in X$ intersection of which is $H : X_a \cap X_b = H$.

Proof.

a) Sufficiency. Let two objects X_a, X_b, such that $X_a \cap X_b = H$, be given. VC H defines on X subset $X_H = \{X_p\}$. Let $\cap X_H = \bigcap_{p=1}^{N_H} X_p = B$.

Since $X_a, X_b \in X_H$, then by definition of intersection $B = H$ and consequently H is an elementary conjunction.

b) Necessity. Let a value combination H, which defines a subset $X_H = \{X_p\}$ on X, be given. Let $\cap X_H = \bigcap_{p=1}^{N_H} = B$. To be an elementary conjunction $(H = B)$, by definition of intersection there should exist at least two objects $X_a, X_b \in X_H$, intersection of which $X_a \cap X_b = H$.

The theorem is proved.

Theorem 3.2. If a value combination H is given, VC H defines a subset $X_H \subseteq X$, $X_H = \{X_p\}$, $X_p \supseteq H$. Then on X_H there always exists an elementary conjunction $A \supseteq H$ such that $|X_A| = |X_H| = N_H$.

Proof. Let $\cap X_H = \bigcap_{p=1}^{N_H} X_p = A \supseteq H$. Since the subset X_H contains all objects $X_p \in X$, for which $X_p \supseteq H$, then by definition of intersection $|X_A| = |X_H| = N_H$ and A is EC.

The theorem is proved.

Lemma 1. If an elementary conjunction is given, then extending it by arbitrary value combination F, $X_{H\&F} \subset X$ ($E_H + E_F \le M$, $H \cap F = \varnothing$) we get a subset of objects $X_{H\&F} \subset XH$ for which $|X_{H\&F}| < |X_H|$.

Proof. Elementary conjunction H defines subset $X_H = \{X_p\} \subseteq X$, $|X_H| = T \le N$, $X_P \supset H$, $p = 1,2,...,T$. Let us separate all objects $X_a \supset F$ from X_H. All X_a correspond to a set $X_{H\&F} = \{X_a\} \subset X_H$, $a = 1,...,|X_{H\&F}| = R$. By definition of an elementary conjunction $X_{H\&F} \subset X_H$ and consequently $R < T$.

The lemma is proved.

Theorem 3.3. To the potency $|X_H|$ of an elementary conjunction H in X there corresponds a maximal number (E_H) of elements h_q in H, $1 \le E_H \le M$ and, inversely, to the number of elements h_q in an elementary conjunction H there corresponds a maximal potency $|X_H|$.

Proof. Let a value combination H be given which defines a subset $X_H \subseteq X$, $X_H = \{X_p\}$, $|X_H| = T$, $p = 1,...,T$, $X_P \supset H$. Let $\cap X_H = \bigcap_{p=1}^{T} X_p = B = \&_q h_q$, $1 \le E_H \le M$, $|X_B| = |X_H| = T$.

If $B \supset H$, then $E_B > E_H$. It ascertains that we can extend a value combination H by a certain value combination F, $H \& F = B$, so that its frequency $|X_{H\&F}|$ will not be decreased ($|X_B| = |X_H| = T$). By the definition of an elementary conjunction (see definition 3.2) B not H is an elementary conjunction.

If $B = H$, it means that H is an elementary conjunction. According to Lemma 1 extending H by arbitrary combination $F = \&_f h_f$ ($1 \le E_F < M$, $E_H + E_F \le M$, $H \cap F = \varnothing$) on X leads to a decrease in potency in relation to elementary conjunction $H: |X_{H\&F}| < |X_H| = T$, $E_B < E_H$.

The theorem is proved.

The Theory of Monotonic Systems

3.3. Main concepts

In this section the main concepts of the theory of monotonic systems are given [3].

Definition 4.1. Let a finite discrete set X, $|X| = N$ and function π_X on it, which maps to each element $\alpha \in X$ a certain nonnegative number (weight) $\pi_X(\alpha)$, be given.

The function π_X is called a weight function if it is defined on any subset $X' \subset X$; the number $\pi_{X'}(\alpha)$ is called a weight of element α on X'.

Definition 4.2. A set X with weight function π_X is called a system (or a system on elements from W) and denoted by $\Pi = (X, \pi_X)$.

Definition 4.3. The system $\Pi' = (X', \pi_{X'})$ where $X' \subset X$ is called a subsystem of the system $\Pi = (X, \pi_X)$.

Definition 4.4. The system $\Pi = (X, \pi_X)$ is called monotonic if in the case of any $\alpha \in X' \setminus \{c\}$, $c \in X'$, $\pi_{X' \setminus \{c\}}(\alpha)$, where X' is any subset of X.

Definition 4.5. Function Q, which is mapping to every subset $X' \subset X$ of monotonic system Π a nonnegative number $Q(X') = \min_{\alpha \in X'} \pi_{X'}(\alpha)$, is called an objective function.

Definition 4.6. The subsystem $\Pi^* = (W, \pi_w)$ of the monotonic system $\Pi = (X, \pi_X)$ in case of which Q-function obtains a maximal value $Q(W) = \max_{X' \subseteq X} Q(X') = \max_{X' \subseteq X} \min_{\alpha \in X'} \pi_{X'}(\alpha)$ is called a kernel of the monotonic system Π: respective $Q(W)$ is called a measure of the kernel quality.

3.4. How to Create a Monotonic System

To use the method of monotonic systems we have to fulfil two conditions.

1) There has to be a weight function $\pi_X(\alpha)$, which will give a measure of influence for every element α of the monotonic system X;
2) There have to be rules f to recompute the weight of the elements of the system in case there is a change in the weight of one element.

These conditions give a lot of freedom to a scientist to choose the weight functions and rules of weight change in the system. The only constraint we have to keep in mind is that the rules f and weight function π have to be compatible in the sense that after eliminating all elements α from the system X the final weights of $\alpha \in X$ must be equal to zero.

4. Solution of the Problem

Having described the concepts required to set up the problem and proved the theorems we can represent our solution.

As shown in Chapter 1, the main disadvantage of the existing algorithms is that we do not know which objects should be intersected among themselves to obtain an original EC. To avoid this at first we should properly arrange the data matrix before we start looking for ECs. The data matrix should be arranged so that its result from intersection can be only an original EC.

It does not mean that data matrix should be rearranged physically, but there is a question: which subset of objects $X' \subset X$ should be intersected with to find an original EC?

To proceed from the theory of monotonic systems we should do the following:

1) find a monotonic weight function, which can obtain its maximal value (or minimum) on the subset of objects with given properties (i.e., which defines an EC);

2) find rules, which can guarantee non-repetition of the EC found (i.e. elimination of this subset)

Further, a description of this process is given.

5.1. ***Creation of a Monotonic System and Extraction of Kernels on*** $X(N,M)$

For creation of a monotonic system on X we calculate for all variables $j=1,...,M$ the corresponding frequencies $|h_j|$, $0 \le |h_j| \le N$. We define the weight function on any subset X' of X as

$$\pi_{X'}(X_{ij}) = |X_{ij}(=h_j)|. \quad (1)$$

Theorem 5.1. The system $\Pi(X',\pi_{X'})$ where $X' \subset X$, $X' = \{X_p\}$, $p=1,...,N'$, $1 \le N' \le N$ and weight function is defined by formula (1), is a monotonic system.

Proof. According to formula (1) weight of each element $X_{pj} = h_j$. At the elimination of any element $X_{et} \in X'' \subset X'$ equals the number of those objects $X_p \in X''$ for which $X_{pj} = h_j$. At the elimination of any element $X_{et} \in X''$ the weight of element X_{pj} may only decrease, i.e.

$$\pi_{X''\setminus\{X_{et}\}}(X_{pj}) \quad (2)$$

for any $X_{pj} \in X''\setminus\{X_{et}\}$.

Supported by definition 4.4 the inequality (2) shows that system $\Pi(X',\pi_{X'})$ is monotonic.

The theorem is proved.

Kernel of the observed monotonic system (according to the definition 4.6) represents such a subset of elements X_{ij} where the minimal weight of the elements is maximal.

According to the theory of monotonic systems kernels are found by iterations [4]: on the basis of maximal and minimal values of the weight function $\pi(X_{ij})$ the threshold value of a weight is computed. If for element X_{ij} its weight $\pi(X_{ij})$ is not higher than the threshold, then it is excluded. In this case the weight for elements $X_{ij} = h_j$ changes. After this process we have separated subset $X' = X \setminus \left\{ X_{ij} \middle| \pi(X_{ij}) \le \text{threshold} \right\} \subset X$.

Now, we have to examine whether X' is a kernel or not. If X' is not a kernel then on X' both $\min\limits_{X_{ij} \in X'} \pi(X_{ij})$ and $\max\limits_{X_{ij} \in X'} \pi(X_{ij})$ are defined and the process of finding the threshold value repeated but now on X'. If the kernel is found we eliminate the kernel W for extracting the next kernels W', $Q(W') \le Q(W)$.

The use of the weight function (1) shows clearly that the labor-consuming and inconvenient iterative process of defining the maximal value of objective function Q (see definition 4.5) can be replaced by:

$$Q(W) = \max_{X' \subseteq X} Q(X') = \max_{X' \subseteq X} \min_{X_{ij} \in X'} \pi_{X'}(X_{ij}) = \max_{X' \subseteq X} \min_{X_{ij} \in X'} \left| X_{ij} \left(= h_j \right) \right| =$$

$$= \min_{X_{ij} \in X'} \max_{X' \subseteq X} \left| X_{ij} \left(= h_j \right) \right| = \max_{X' \subseteq X} \left| X_{ij} \left(= h_j \right) \right|.$$

In this case there are no contradiction to the theory of monotonic systems, since $\pi_{X' \setminus \left\{ X_{ij} \middle| \pi(X_{ij}) \le \text{threshold} \right\}}(X_{ij}) \le \pi_{X'}(X_{ij})$.

The above described weight function (1) and the process of extracting of kernels are used in algorithm MONSA (MONotonic System Algorithm).

5.2. Algorithm MONSA

By essence MONS is recursive algorithm. We estimate its backtracking version here.

In this algorithm the following denotations are used:

t – the number of the step (or level) of recursion,

FT_t – frequency table for a set X_t,

$INTERSECTION_t$ – vector of elements of intersection over set X_t (i.e. EC)

Init – activity for initial evaluation,

$\mathrm{1Eliminate}(t+1)$ – activity that prohibits arbitrary output repetition of already separated elementary conjunction on level $t+1$,

$\mathrm{2Eliminate}(t+1)$ – activity that does not allow the output of the separated elementary conjunction on the same (current) level $t+1$ and on steps $t, t-1, ..., 0$.

Algorithm MONSA.

Init

$t = 0$, $\mathrm{INTERSECTION}_0 = \{\ \}$

To find a table of frequencies FT_0 for all attributes in X_0.

Do while there exists $\mathrm{FT}_s \# \varnothing$ in $\{\mathrm{FT}_s\}$, $s \le t$

For an element $h_f \in \mathrm{FT}_t$ with frequency $V = \max \mathrm{FT}_t(h_f) \# 0$

Do

To separate submatrix $X_{t+1} \subset X_t$ such that

$$X_{t+1} = \left\{ X_{ij} \in X_t \,\middle|\, X.f = h_f \right\}$$

To find a table of frequencies on X_{t+1}

If there exist on X_{t+1} hu, $1 \le u \le M$, such that

[$hu \in \mathrm{INTERSECTION}_{t+1}$ and $\mathrm{FT}_{t+1}(hu) = 0$ and frequency of hu in $X_{t+1} = V$] then go to BACK (we have already extracted the set X_{t+1} at least once)

$\mathrm{1Eliminate}(t+1)$

Add elements j with $\mathrm{FT}_{t+1}(j) = V$ to $\mathrm{INTERSECTION}_{t+1}$

$\mathrm{2Eliminate}(t+1)$

If there exist attributes to analyze, then $t = t+1$

Endif

Output of $\mathrm{INTERSECTION}_t$

$\text{INTERSECTION}_{t+1} \leftarrow \text{INTERSECTION}_t$

Endfor

BACK: $t = t - 1$

$\text{INTERSECTION}_{t+1} \leftarrow \text{INTERSECTION}_t$

Enddo

ALL INTERSECTIONS ARE FOUND

END: end of algorithm.

Further, we shall add some comments to explain the work of the algorithm.

The main idea of the work of the algorithm MONSA is simple:

1) subset $X_{t+1} \subset X_t$ of objects with certain properties is being separated,
2) then intersection over this subset X_t is being found.

We repeat these two steps until we have found all subsets X_t, $t = 0,1,...,U$, $0 < U \le M$.

In the case of algorithm MONSA the following methods are used. They differ from classical algorithms:

1) subset $X_{t+1} \subset X_t$ is defined by the function that obtains on X_t its maximal value,
2) to find an intersection over the set X_{t+1} the frequency table FT_t of the set X_t can effectively be used as follows:

maximal frequency $MAX = |X_{t+1}|$ of a certain element $X_{ij} \in X_{t+1}$ in frequency table FT_{t+1} is defined by the frequency in FT_t of EC, which was the base to separate X_{t+1}. Consequently, all elements $X_{ij}(= h_j) \in X_{t+1}$ the frequencies of which equal MAX, appear simultaneously in all objects of X_{t+1} and define an intersection over the set X_{t+1}.

According to the theory of monotonic systems we interpret the above described process as follows:

In the working process of MONSA those subsets $X_{t+1} \subset X_t$, $t = 0,1,...,U$, $0 < U \leq M$, are found, on which objective function Q obtains its maximal value related to X_t. Thereupon set X_{t+1} is being eliminated from further analysis and the whole procedure that is described above is being repeated once again until all subsets X_{t+1}, $t = 0,1,...,M-1$ are found.

Elimination of set X_{t+1} creates conditions for maximization of objective function Q on set $X_t \setminus X_{t+1}$. Elimination guarantees non-repetition of set X_{t+1} separation.

The process of elimination represents prohibition of the separated set X_{t+1}. It happens in two ways:

1) to avoid repetition of a separation of the already separated subset X_s from X_{t+1} we have to keep in mind their descriptions. In the algorithm MONSA it is realized by using a simple technique – in set X_{t+1} frequency table FT_{t+1} we nullify those cells with non-zero values the value of which in set X_t frequency table FT_t is equal to zero (in MONSA the activity 1Eliminate);

2) if in FT_{t+1} the frequency of some attribute value equals its frequency in frequency table FT_t, it means that we may prohibit for it the corresponding subset X_m, $m < t$, since intersection over X_m represents the same intersection as over the given X_{t+1}, i.e. $X_{t+1} = X_m$ does. In the algorithm MONSA this process goes on by nullifying the corresponding content in FT_t (in MONSA activity 2Eliminate). As we move zeros from FT_t to the FT_{t+1} (activity 1Eliminate), then it guarantees that we shall not separate the other set $X_m = X_{t+1}$.

In the following we shall prove the theorem that confirms our assertion.

Theorem 5.2. Nullifying the frequency of value categories, which describe set $X_{t+1} \subset X_t$ in the frequency table, FT_t does not allow repetitive separation of X_{t+1} from set X_t.

Proof. According to the theory of monotonic systems a certain element $X_{ij} \in X$ is considered to be eliminated if its weight $\pi(X_{ij}) = 0$, i.e., it does not participate in the further analysis. Since the weight function in the algorithm MONSA is the frequency of attribute's value, then nullifying the frequency in the frequency table FT_t of attributes, which describes the set X_{t+1}, means that their weight in set X_t $\pi(X_{ij}) = 0$ and for that reason repetition of set X_{t+1} separation from X_t is impossible.

The theorem is proved.

Theorem 5.3. Nullifying of the frequency of such attributes value categories with non-zero frequency in the frequency table FT_{t+1} (weight in set X_{t+1}), whose frequency in frequency table FT_t (weight in set X_t) equals zero, does not allow separation of the subsets X_{t+2} described by them from the subset X_{t+1}.

Proof. According to the theory of monotonic systems a certain element $X_{ij} \in X$ is considered to be eliminated if its weight $\pi(X_{ij}) = 0$, i.e., it does not participate in the further analysis. Since the weight function in the algorithm MONSA is the frequency of attribute's value, then nullifying of frequency in the frequency table FT_{t+1} of attributes, which have zero frequency in FT_t means that their weight on set X_{t+1} $\pi(X_{ij}) = 0$ and for that reason repetition separation of sets described by them from X_{t+1} is impossible.

The theorem is proved.

Theorem 5.4. If the frequency of a value category of a certain attribute in set X_{t+1} equals its frequency in a certain set $X_m \supset X_{t+1}$, $0 \le t$, then we have to prohibit this value category in the set X_t.

Proof. Set X_{t+1} is eliminated using the frequency table FT_t and all subsets $X_{t+1} \subset X_t$ are considered to be separated from set X_t if $FT_t = \varnothing$. In the case of the frequency of certain attribute value category in FT_t (weight in subset X_{t+1}) equals to its frequency in the frequency table FT_m (weight in subset X_m), $X_m \supset X_t$, $m < t$, then it means that we may repeatedly separate the subset $X_{m+1} = X_{t+1}$, from X_m. To avoid this situation we should eliminate these attributes value categories from all subsets X_m, $0 \le m \le t$, $X_m \supset X_{t+1}$.

The theorem is proved.

In the following we will further explain algorithm MONSA.

1. MONSA is a back-tracking algorithm. It forms a hierarchical grouping tree (HGT). Only one branch of the tree is currently under consideration.
2. First, we form a HGT branch from a root node, for which attribute's value frequency is the greatest, to the leaf, i.e. a chain of nodes with the length U, $1 < U \le M$. At that, levels $t = 1,2,...,U$ are passed, $U \le M$ because we can add more than one new element h_q (see definition 3.3 of maximal EC) to EC of the previous level, i.e. EC is being formed not by one attribute only. To the new EC at the level t one can add at the same time $U - t$ different attribute values.
3. At each level $t+1$ there takes place a separation of a new subset X_{t+1} from the previous subset X_t of objects defined by the attribute value category with the greatest frequency on the subset X_t.
4. After obtaining an EC from M elements we repeat the process of EC extracting until the frequency table of the last level is exhausted (all cells are nulled). In this way we have separated all EC related to EC of the previous level proceeding from order of accumulation of attributes values to EC.

5. The cell content that belongs to the frequency table of the previous level is nullified only if the frequency table, which corresponds to the subset of objects described by the corresponding attribute value is completely exhausted. Zero in a certain cell in the given subset all its following ECs (proceeding from order of accumulation of attributes) are separated. Nullifying correspondents to the algorithm's MONSA activity ELIMINATE.
6. The work of an algorithm ends at level "null", which signifies that set's $X(N,M)$ frequency table is exhausted, i.e. all EC on set X are separated.

5.3. Theorem of X^H *and Kernel Identity*

To prove the described discussion we estimate the following theorem.

Theorem 5.5. The subset $X^H \subseteq X$ extracted by the algorithm MONSA is a kernel $\Pi(X_H, \pi_{X_H})$ of the system $\Pi(X, \pi_X)$.

Proof. The algorithm MONSA extracts ECs on set X. Every EC H describes set $X^H \subseteq X$, $X^H = \{X_{ij} \in X \mid X_{ij} \in H, X_i \supseteq H\}$. At the same time $X^H \subseteq H$, $X_H = \{X_i \in X \mid X_i \supseteq H\}$. According to theorem 3.3 EC H has maximal frequency on set X_H. All elements $h_q \in H$ have equal frequency $|X_H|$ on set X_H. In terms of theory of monotonic systems we have built a monotonic system on X_H with threshold value $\text{THRESHOLD} = \max_{X_{ij} \in X_H} \pi(X_{ij}) = \max_{X_H \subseteq H} |h_j (= X_{ij})| = |X_H|$. If we eliminate elements $X_{ij} \in X_H$ the weight of which $\pi(X_{ij}) << \text{THRESHOLD}$, then remaining elements form a set $X^H = \{X_{ij} \mid X_{ij} \geq \text{THRESHOLD}\}$, where $\max \pi(X_{ij}) = \min \pi(X_{ij})$ and on which the objective function obtains its greatest value related to X_H, consequently X^H is a kernel $\Pi(X_H, \pi_{X_H})$ of the system $\Pi(X, \pi_X)$.

The theorem is proved.

5.4. Estimation of Algorithm Complexity

The above-described algorithm does not appear so compact as the one presented in Chapter 1. Actually MONSA work consists of repetitive formation of frequency tables. Here it is not necessary to go through the whole data matrix to find a frequency table, but it is enough to go through those attributes in objects with certain properties that prevent EC repetition.

Here we represent the theorem, which proves the complexity of an algorithm for a number of objects $N = K^M$. The fast estimates of MONSA for the rest of the cases are given in Chapter 6.

5.4.1. Theorem of Algorithm MONSA Complexity

Theorem 5.6. Suppose that a finite discrete data matrix $X(N,M)$ is given, where $N = K^M$ and each element X_{ij} can obtain a value in the interval $h_j = 0,1,2,...,K-1$. Then the complexity of the algorithm MONSA to find all $(K+1)^M$ elementary conjunctions is $O(N^2)$ operations.

Proof. Algorithm MONSA forms a hierarchical grouping tree (HGT), the nodes of which are ECs. At each level t, $1 \le t \le M$ the frequency table FT_t is formed on set X_t, $|X_t| = N/K^t$. Level t is considered to be finished in relation to set X_t if the whole frequency table $FT_t = \varnothing$ (i.e. all nodes of HGT, which start from node t are passed trough). To avoid their repeatedly passing to nodes $t+1$ of HGT in algorithm MONSA, they are eliminated (see theorems 5.2 to 5.4).

Thus, at certain passing of level t is under consideration of N/K^t objects of set X. At the given order of passing of attributes the number of those noncovered sets X_t is equal to K^t, which we can form in level t related to the set X. When forming FT_t for each set X_t on level t $U-t$ attributes, $t < U \le M$, were obtained.

Consequently, for the whole level t with the given order of attributes we have made $(U-t)\cdot N \cdot K^t / K^t = (U-t)\cdot N$ operations on set $X(N,M)$.

Since X_t cannot be intersected, i.e. subsets X_t that correspond to the nodes of HGT level t are not intersected themselves, and in algorithm MONSA these subsets X_t are eliminated (see theorems 5.2 to 5.4), then in relation to the whole X we may consider separation of EC as a process of combination of attributes by $1,2,...,M$. In this process all EC on $X(N,M)$ are extracted for every attribute combinations. Total number of attribute combinations on $X(N,M)$ is $2^M - 1$. Since for the generation of a new attribute combination it is necessary to form a frequency table FT for $U - t$ vacant attributes in level t.

In MONSA to form FT as many attributes as there are steps (levels) to the leaf U are obtained, thus a chain with the length of $U - t$. In the case of M attributes the number of those chains with length of M attributes is one, $(M-2)$ attributes -2, $(M-2)-4$, $(M-2)-8,\ldots,M-(M-1)$ attributes -2^{M-1} (see Table 1).

By all levels t we have formed K^t noncovered subsets X_t, in each subset there are N/K^t objects. Likewise for each subset X_t the chain is passed with $U - t$ attribute length to create a frequency table. This, for the formation of all frequency tables we have passed $2^M - 1$ chains or, in other words $2\cdot(2^M - 1 - M) = \sum_{t=1}^{M} 2^t - 1$ attributes. For extracting $(K+1)^M$ EC we have carried out $2\cdot(2^M - 1 - M)\cdot N$ operations or, in other words, we have passed $X(N,M)$ $\{2^{M+1} - (2\cdot M + 2)\}/M$ times.

Table 1. Number of chains passed by algorithm MONSA

Length of chain (attr)	Number of chains	Attributes In total
M	1	M
$M-1$	2	$2\cdot(M-1)$
$M-2$	4	$2\cdot(M-1)$
$M-3$	8	$2\cdot(M-1)$
…	…	…
$M-(M-1)$	2^{M-1}	2^{M-1}
	$\sum = 2^M - 1$	$\sum = 2\cdot(2^M - 1 - M)$

As we can see, the complexity of algorithm MONSA for $N = K^M$ does not depend on the number of attribute value categories K. As the minimum number of value categories K is equal to two, it means that in case $K = 2$, $N = K^M = 2^M$ and the complexity is

$$2 \cdot (2^M - 1 - M) \cdot N = 2 \cdot (N - 1 - M) \cdot N = O(N^2)$$

operations. If $K > 2$, then $N = K^M > (2^M - 1 - M)$ and complexity of MONSA is decreasing relative to N.

The theorem is proved.

6. Discussion

At first sight one may think that the complexity of the algorithms is $O((K \cdot M) \cdot (K \cdot (M-1)) \cdot (K \cdot (M-2) \cdot \ldots \cdot K)) = O(K^M \cdot M!)$ operations in case of arbitrary N, because it appears as a frequency table from $M - t$ attributes is formed for every frequency table element whose frequency is not equal to zero. Such a judgment would be justified if we had tested all possible permutations of attributes. But in reality those of them are tested on which subsets X_s defined by corresponding to elementary conjunction do not overlap, i.e. $X_s \# X_f$, $s \# f$, where s, f are elementary conjunctions.

Theorem 5.6 has proved algorithm MONSA complexity $O(N^2)$ operations for $N = K^M$ objects. It is the case when the number of EC is maximal: $L = (K+1)^M$. But in real life such data matrixes are possible only when M and K are very small. We come to the question here of how to estimate the computational complexity in the case $N < K^M$?

In the following we will suggest some practical criteria which serve as basis for estimation of the complexity depending on N and M. We shall use the same denotation as in Chapter 1: $X(N, M)$ is a finite discrete data matrix, N is the number of objects, M – that of attributes, K – that of attribute categories, L – that of VC on set X.

First, we will demonstrate a simple connection, which allows us to estimate the number of VC depending on N and M: assuming that

$$\frac{K^M}{N} \approx \frac{(K+1)^M}{L}, \text{ we obtain}$$

$$L = \frac{N \cdot (K+1)^M}{K^M} = N \cdot (1+1/K)^M,$$

where $(1+1/K)^M$ is the mean number of value combinations per object. For instance, if $K=2$, $M=10$, $N=100$, then $L = 100 \cdot 57{,}665 \approx 5767$ VCs.

Further we will try to estimate the computational complexity depending on N and M. It is clear that this complexity is not decreased because of N-value only, but the reason for it is that by decreasing N, we also decrease the number of formatted frequency tables and the number of empty cells in frequency tables is in creased.

For instance, if $K=2$, $M=9$, $N=2$, then by the complexity formula we should pass the full data matrix $(1024-20)/9 = 1024/9$ times, but actually there are less than 2 passages resulting in the extraction of three EC: two objects and their intersection. Overestimates are caused by the fact that formula considers only the worst case by which formation of frequency tables is carried out for all attribute combinations (by $1,2,...M$) needs. Actually only three frequency tables have to be formatted for $N=2$.

The precise formula of computational complexity may be as follows: $\sum_{f=1}^{F} \sum_{p=1}^{(M-K-t)} N_p$ operations, where F is the number of formatted frequency tables on set X, t is the number of empty sells in the frequency table FT_f, N_p is the absolute number in a cell of the frequency table (frequency of attribute value).

It should be noted that in this formula t is a variable. Its value is increasing in the working process when sets X_H are eliminated.

7. Conclusion

The basic algorithm MONSA, which is described in this paper appears to be very efficient and has found application in various systems [5, 6, 7]. As compared to the algorithms in Chapter 1 this algorithm allows to obtain very fast results [7]. The disadvantages uncounted with algorithms B1, B2 and B3 are not the case here and if additional criteria are skillfully set up optimization is easily obtainable.

References

1. Garey, M., Johnson, D. '*A guide to the Theory of Np-Completeness*.' San Francisco: W.H. Freeman and Company, 1979.
2. Zabezhailo, M., Ivashko, V., Kuznetsov, S., and others. 'Algorithmic and Programming Tools for DSM-Method for Automatical Generation of Hypotheses.' Scientific-Technical Information, Seria 2, 1987, No 10, pp. 1-14 (in Russian).
3. Mullat, J. 'Extremal Subsystems of Monotonic Systems.' Automation and Remote Control, 1976, No 5, pp. 130-139; No 8, pp. 169-178 (in Russian), http://www.datalaundering.com/download/extrem01.pdf, http://www.datalaundering.com/download/extrem02.pdf.
4. Mullat, J., Vyhandu, L. 'Monotonic Systems in Scene Analysis.' *Symposium: Mathematical Processing Methods for Data Analysis and Processing of Cartographical Data*. Tallinn, 1979, pp. 63-66, http://www.datalaundering.com/download/cardgraf.pdf.
5. Kuusik, R. 'Generator of Hypotheses for Qualitative Data.' *Trans. of Tallinn Tech. Univ*., 1987, No 645, pp. 141-148 (in Russian).
6. Kuusik, R. 'Application of the Theory of Monotonic Systems for Decision Trees Generation.' *Trans. of Tallinn Tech. Univ*., 1989, No 705, pp. 47-58.
7. Roosman, P. 'A new Method for Learning from Examples.' Computers and Data Processing, Seria XVIII, 1991, No 8, pp. 31-51 (in Estonian).

Kiire hierarhilise klasteriseerimise algoritm ja monotoonsete süsteemide teooria

Kokkuvõte

Artiklis esitatakse monotoonsete süsteemide teorial baseeruv hierarhiliste grupeerimise algoritm, näidatakse tema eelised teiste sama liiki algoritmide suhtes. Tõestatakse, et $N \cdot M$ andmetabeli korral, kus N – objektide arv, M – tunnuste arv ning iga tunnus j, $j = 1,...,M$, võib omada diskreetseid väärtusi vahemikus $h_j = 0,1,...,K-1$, juhul, kui $N = K^M$, on algoritmi keerukuseks $O(N^2)$. Seejuures eraldatakse kõik $(K+1)^M$ klastrit.