ASSOCIATION RULE MINING WITH N-DIMENSIONAL UNIT CUBE CHAIN SPLIT TECHNIQUE

Levon Aslanyan, Robert Khachatryan

Abstract: *This paper considers the association rule mining problem and provides an alternative to APRIORI algorithm, for solving this problem by splitting up the n-dimensional unit cube vertices into the chains using the well know technique introduced by G. Hansel. We use the further development of this technique given by G. Tonyan, which adds computability on levels of chains and chain splits. A brief description of the software implementation of the introduced alternative approach is given.*

Keywords: *association rule, data mining, unit cube.*

ACM Classification Keywords: *1.5. Pattern recognition, H.2.8 Database applications, Data mining.*

Introduction

Association rule mining problem is one of the main objectives of "data mining" research discipline – /finding the knowledge and regularities in large amounts of experimental data sets/. Given a set $I = \{x_1,...,x_n\}$, consisting of n various elements (items) x_i. Subsets of items (itemset) $X \subseteq I$ are considered, and we say that it is given a k-itemset, when $|X| = k$. Let D is a database of records (transactions) that are subsets of the elements (in this view records are lists of elements, but the same information can be given equivalently by the characteristic 0-1 n-vector of itemset X in I). We assume that the records can be in repetition (considering a multiset), but they are provided with an additional field, which composes the key in the database.

We say that a record $T \in D$ is contributing to set of elements X, if $X \subseteq T$. Association rule is an "if-then" type rule $X \Rightarrow Y$, the fulfillment of which is related with certain conditions. Let X, Y be sets of elements where $X \cap Y = 0$. The ratio of number of all records of D contributing to X, and the number of records of D - is called support of X in D,

$$\sup p(X) = |\{T \in D, X \subseteq T\}| / |D|.$$

Next to this is the concept of support for a rule $X \Rightarrow Y$:

$$\sup p(X \Rightarrow Y) = \sup p(X \cup Y).$$

Another important property of rules is the confidence that is defined as:

$$conf(X \Rightarrow Y) = \sup p(X \cup Y) / \sup p(X),$$

which is the conditional probability that a record contains Y when it is known that it contains X. Practical implementation of association rule mining problem is a subject of intense theoretical and algorithmic studies. It is well known that the problem splits naturally into two stages [AGRAW, 1996]. The first step is the construction of frequent fragments, those that occur in the database with frequencies above the predetermined value of support. The second stage is actually the phase of synthesizing the rules of given confidence, from the set of frequent subsets constructed during the first stage of algorithm.

The most accepted algorithm for synthesis of association rules is APRIORI [AGRAW, 1996]. It builds the set of frequent subsets with a so-called building up method. APRIORI considers one-element subsets, and for them by one run on the database computes their frequencies. Next, it considers all two-element subsets, one-element subsets of which are frequent, and verifies their occurrence in the table. Thus the frequent subsets can be building up to the state when it is including subsets that are not frequent enough. Computational complexity is significant, it is especially important because algorithm must be used on very large data volumes.

Are there any alternative approaches for building rules? There is a huge number of approaches, ideas and algorithm that address this issue. In this paper we propose one new approach, which connects the well-known results from the geometry of the n-dimensional unit cube to the problem of algorithmic generation of association rules with given threshold for rule support and confidence.

A brief characterization of this approach is as follows. n-dimensional unit cube B_n is a regular lattice consisting of 2^n vertices that correspond to binary strings of length *n*, which are usually arranged in layers in the way that on the *k*-th layer there are all those vertices that have *k* units (1 values). Vertices that differ in one coordinate are called adjacent and are connected by an edge. Chain in B_n is a sequence of adjacent vertices. A chain is called growing if it contains at most one vertex in one layer.

G. Hansel showed that B_n can be split into growing chains under certain conditions. Further, he considered the monotone Boolean functions and built an algorithm of optimal recognition of these functions using the constructed chains. Relationship of these constructions with the association rules are that frequent subsets with given parameters correspond to a set of zero value vertices of a monotone Boolean function.

Direct use of this technique of Boolean function recognition is difficult because the constructing and storing the Hansel chains is a problem of algorithmic exponential complexity – in computation, and in memory used.

G. Tonoyan was able to offer a computational approach to the work with chains. This is fundamentally and significantly simplifying the recognition algorithm although the reminding complexity is still very high. The idea is in selecting one particular chain split in the collection of Hansel splits. Then a number of functions are introduced that map chains and their elements to each other. In total, this provides the necessary information to recognize

monotone Boolean functions and eliminates the need in storing the complete structure of Hansel chains. This means sensitive economy of memory versus a small additional computation over tye chain split.

The total aim of this paper is to introduce the necessary chain split and computation technique in terms of problems of search of association rules in large databases. Additionally, it is to take into account one more important feature of the problem for mining association rules. It is known that in data mining the number of considered elements, *n,* - is very large. It is also know that frequent subsets consist of relatively small number of elements. According to this an assumption occurs that there exists a value *k* such that all subsets above this power are not frequent. It turns out that the problem of search of frequent subsets is equivalent to decoding of a special class of monotone Boolean functions, which in turn requires an expansion of the results mentioned above for general Boolean functions, according to some restrictions of the set of functions considered. Extended results are introduced in terms of problem of frequent subsets synthesizing, thus providing the way of determining the set of all maximal (largest by inclusion) frequent subsets, without considering and constructing their sub-subsets. This avoids the part that particularly complicates the building up process.

On geometry of the n-dimensional unit cube

Variable with the only values 0 and 1 (false and true) is called a Boolean variable. n-dimensional Boolean function is a single-valued transformation of the set of all vectors composed by n Boolean variables on to the Boolean set $B = \{0,1\}$. The domain where the Boolean function is given is known as the set of vertices of the n dimensional unite cube B_n that is n-th Cartesian degree of set B. B_n is the set of all binary vectors $\tilde{\alpha} = (\alpha_1,...,\alpha_n)$, which are called vertices or points. Usually, B_n also includes a certain structure, a graph, in which vertices of B_n are placed in horizontal layers, a layer contains all the vertices with a given number of *k* ones, $1 \le k \le n$, and the layers are arranged vertically, starting from the zero layer (at the bottom) to the layer with number n. Layer *k* consists of C_n^k vertices. Two vertices $\overline{\alpha}$ and $\overline{\beta}$ are called adjacent if they differ in exactly one coordinate. These neighboring vertices are connected by an edge in the graph structure of B_n.

Vertices of B_n are organized as follows: a point $\tilde{\alpha} = (\alpha_1,...,\alpha_n)$ of B_n precedes the point $\tilde{\delta} = (\delta_1,...,\delta_n)$ of B_n, if $\alpha_i \le \delta_i$, $1 \le i \le n$. The fact that a point $\tilde{\alpha}$ precedes the point $\tilde{\delta}$ is denoted by $\tilde{\alpha} \le \tilde{\delta}$. If $\tilde{\alpha} \le \tilde{\delta}$ and $\tilde{\alpha} \ne \tilde{\delta}$ then we write $\tilde{\alpha} \prec \tilde{\delta}$. Two different points $\tilde{\alpha}$ and $\tilde{\delta}$ are called comparable if one of the following conditions is true: $\tilde{\alpha} \prec \tilde{\delta}$ or $\tilde{\alpha} \succ \tilde{\delta}$.

It is evident, in general, that to uniquely identify a Boolean function it is necessary to know its values at all points of the n-dimensional unit cube. If the function belongs to some specific class that is narrower than the set of all Boolean functions, then for the unique determination of its values at all points of B_n is not necessary to know in advance the values of function at all points of B_n, and sometimes it is enough to know the values on a subset of B_n. For example, to uniquely identify a symmetric Boolean function of n variables (these functions possess the

same value on each layer of B_n) it is enough to know its values on the set of points from B_n which is intersecting with all layers of B_n .

Boolean function $f(x_1, x_2, ..., x_n)$ is called monotone if from the fact that $\tilde{\alpha} \prec \tilde{\delta}$ it implies that $f(\alpha_1, \alpha_2, ..., \alpha_n) \leq f(\delta_1, \delta_2, ..., \delta_n)$. The class of all monotone Boolean functions of n variables is denoted by M_n . Some geometric properties of monotone Boolean functions are evident. To each function there is a unique set $\hat{f}^0$ of incomparable vertices of B_n , so that $f(\tilde{\alpha}) = 0$ iff $\tilde{\alpha}$ precedes one of these points. Geometrically the area is a union of sbcubes, composed by vetex $\tilde{0}$ and the vertices of $\hat{f}^0$. Another important property is that on growing chains of vertices the function values - 0's and 1's fill two different intervals at most.

Two tasks of recognition about the monotone Boolean functions are rising in different applications. One is the recognition whether $f(\tilde{x}) \in \mathrm{M}_n$, and the second is in deciphering of $f(\tilde{x})$ given that $f(\tilde{x}) \in \mathrm{M}_n$. We address the second topic because of it's identity to the problem of frequent itemset mining.

Suppose that an arbitrary (unknown to us) function $f(\tilde{x}) \in \mathrm{M}_n$ is given by an operator A_f , which gives the value $f(\tilde{\alpha})$ by the given input $\tilde{\alpha} \in B_n$. Given the operator A_f it is required to fully restore the set of values of the function $f(\tilde{x})$. After each call to the operator which resumes the value $f(\tilde{\alpha})$ for the point $(\alpha_1, ..., \alpha_n) \in B_n$ other points of B_n become determined through the monotony property. It is clear that we should strive for optimality of these algorithms that is to minimize the steps of applying to A_f .

Consider the set R of all algorithms that solve this problem. That is, for a monotone Boolean function $f(x_1, x_2, ..., x_n)$ an algorithm from R exploiting the operator A_f restores the complete table of values of $f(\tilde{x})$. Obviously the work of algorithms consist of several stages. Algorithm selects a point $\tilde{\alpha} \in B_n$ and with help of operator A_f computes the value $f(\alpha_1, \alpha_2, ..., \alpha_n)$ (selection). The resulting value of the function at the point $\tilde{\alpha}$ is inserted into the table of computed values of the function. The table is extended by monotonity, which includes determination of all points that can't have 0 or 1 values arbitrarily after knowing the value at $\tilde{\alpha}$ (extension). For example if $f(\tilde{\alpha}) = 1$ then for all points $\tilde{\delta}$ that are higher that $\tilde{\alpha}$ (according to the order of vertices defined above) $f(\tilde{\delta}) = 1$ and the table of values of f is filled in accordingly. Next step is the rule that selects another input for operator A_f and the table of values of f is filled again by monotonity. This process is repeated until the table of values is filled completely.

Obviously a pair <algorithm $r \in R$ and monotone function $f(x_1, x_2, ..., x_n)$ > can be associated with a number $\varphi(r, f)$ that is the number of calls to the operator A_f during recovery of table of values of function $f(x_1, x_2, ..., x_n)$ by the algorithm r .

It is appropriate to evaluate the quality of the algorithms R using function $\varphi(R,f)=\min_r \varphi(r,f)$. We have a condition, $f \in M_n$. The complexity of recognition of class of n-dimensional monotone functions can be characterized by function $\varphi(n)=\varphi(R,M_n)=\max_f \varphi(R,f)$, where the maximum is taken over all monotone functions.

Let us introduce some general terms on function deciphering [KOR,1965]. Suppose we are given a certain class N of Boolean functions and a function f, belonging to this class. The set of points $G(f,N)$ from B_n is called resolving set for the pair (f,N), if from the fact that

the function g belongs to *N*,

values of f and g are the same on the set $G(f,N)$

it follows that $f=g$.

To restore the table of values of a functions it is sufficient to determine the values of function on some of its resolving sets. Resolving set $G(f,N)$ is called a deadlock resolving set for (f,N), if no subset of it is resolving for the pair (f,N).

Let us denote by $H(\tilde{\alpha})$ the set of points $\tilde{\delta}$ satisfying the condition $\tilde{\delta} \succ \tilde{\alpha}$, and by $L(\tilde{\alpha})$ - the set of points $\tilde{\gamma}$ such that $\tilde{\gamma} \prec \tilde{\alpha}$.

The upper zero of monotone function $f(x_1,x_2,...,x_n)$ is the point $\tilde{\alpha}$ from B_n such that $f(\tilde{\alpha})=0$ and $f(\tilde{\delta})=1$ for all points $\tilde{\delta} \in H(\tilde{\alpha})$.

The lower one of a monotone function $f(x_1,x_2,...,x_n)$ is a point $\tilde{\alpha}$ such that $f(\tilde{\alpha})=1$ and $f(\tilde{\gamma})=0$ for any point $\tilde{\gamma} \in L(\tilde{\alpha})$.

Let $Z(f)$ denotes the set of all upper zeros of a monotone function $f(x_1,x_2,...,x_n)$, and $O(f)$, - the set of all lower ones. Each monotone Boolean function has a unique deadlock resolving set that is included in its all resolving sets (mention that this is not the case for other classes, fir instance in class of simmetric Boolean functions that we mentioned above above). This deadlock resolving set for a monotone Boolean function is the set $G(f)=Z(f)\cup O(f)$.

V. Kororbkov has obtained the following result concerning the upper and lower grades of $\varphi(n)$.

Korobkov's Theorem

$C_n^{[n/2]}+C_n^{[n/2]+1} \le \varphi(n) \le bC_n^{[n/2]}(1+\varepsilon_n)$ where $b=\frac{8}{(\sqrt[3]{16}-1)^{3/2}}$ and $\varepsilon_n \to 0$ when $n \to \infty$.

The proof of theorem uses the logic algebra function $h(x_1, x_2, ..., x_n)$ defined as:

$$h(\alpha_1, \alpha_2, ..., \alpha_n) = \begin{cases} 1, if\ [n/2]+1 \le \sum_{i=1}^{n} \alpha_i \le n \\ 0, if\ 0 \le \sum_{i=1}^{n} \alpha_i \le [n/2] \end{cases}$$

It's obvious that in this case $G(h)$ has exactly $C_n^{[n/2]} + C_n^{[n/2]+1}$ points.

Chain split in monotone recognition

Let us give the definition of increasing chain and the property of relative supplement:

1. Increasing chain in the structure of B_n, is a sequence $\tilde{\beta}_1, \tilde{\beta}_2, ..., \tilde{\beta}_k$ of elements of B_n such that $\tilde{\beta}_{i+1}$ is obtained from $\tilde{\beta}_i$ with replacement of one zero in the set of coordinates to the one.

2. Suppose we are given three elements $\tilde{\alpha}_1 \prec \tilde{\alpha}_2 \prec \tilde{\alpha}_3$ forming an increasing chain. Relative supplement of $\tilde{\alpha}_2$ on the interval $[\tilde{\alpha}_1, \tilde{\alpha}_3]$ is the fourth element $\tilde{\beta}$, which forms together with $\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3$ a two-dimensional subcube (see Figure 1).

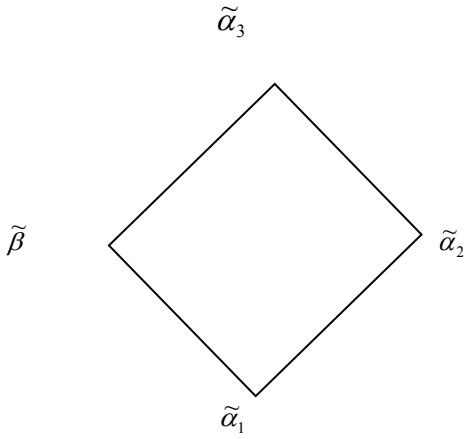

Figure 1

Hansel's Lemma

n dimensional unit cube B_n, endowed with the usual order relation can be covered with a set of $C_n^{[n/2]}$ disjoint increasing chains satisfying the following properties:

- the number of chains of length $n-2p+1$ is equal to $C_n^p - C_n^{p-1}$, where $0 \le p \le [n/2]$. Minimal element of each such chain is a point with p units and $n-p$ zeros and the maximal with p zeros and $n-p$ ones.
- given three elements $\widetilde{\alpha}_1 \prec \widetilde{\alpha}_2 \prec \widetilde{\alpha}_3$ that form an increasing subchain placed on some chain of length $n-2p+1$, then the relative supplement $\widetilde{\alpha}_2$ on the interval $[\widetilde{\alpha}_1, \widetilde{\alpha}_3]$ belongs to a chain of length $n-2p-1$.

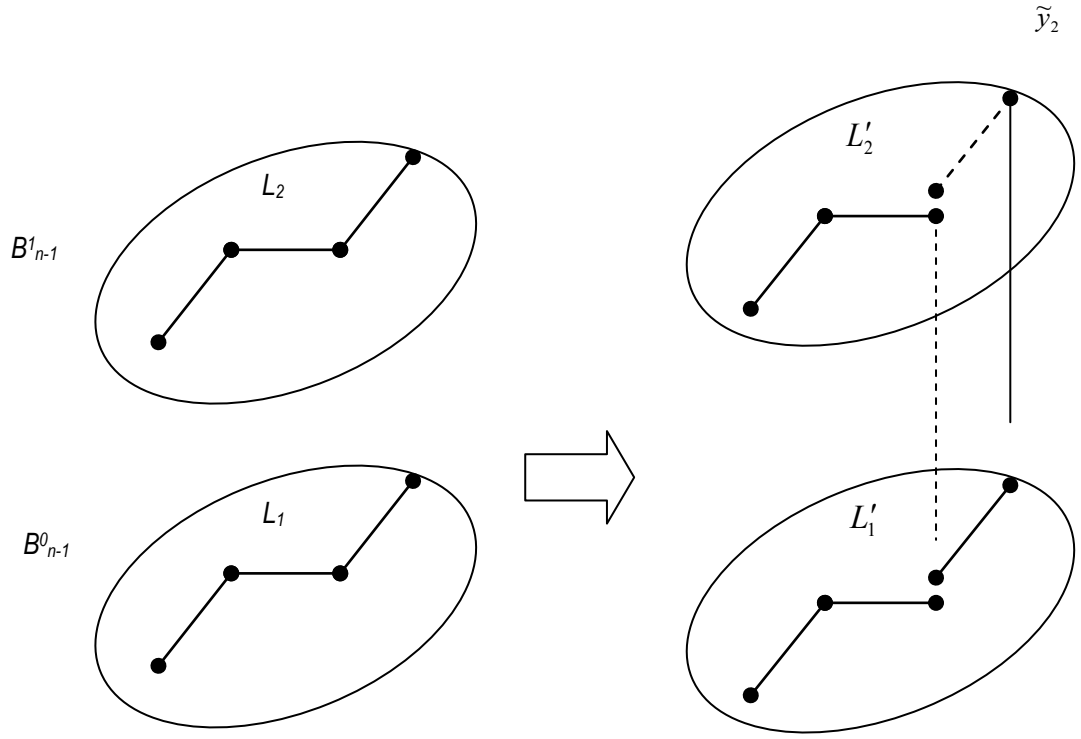

Figure 2.

The proof of this fact is inductive by n. Figure 2 clarifies how the greatest element $\widetilde{y}_2$ of chain L_2 is removed from the L_2 and is added to chain L_1, becoming its new greatest element.

Theorem ([HANS, 1966]**).** Minimal number of operations $\varphi(n)$ of monotone Boolean function's recognition algorithm's is equal to:

$$\varphi(n) = C_n^{[n/2]} + C_n^{[n/2]+1}.$$

Now we bring some more information about the chain computations for the restricted Boolean functions. Let R_n denotes the set of all chains built by and satisfying the conditions 1 and 2 of Hansel's lemma. Consider an element of R_n .The set of all length l chains denote by $[l]_n$ .The optimal algorithm for recovering a monotone Boolean function is denoted by F_0 . The description of F_0 is as follows.

Operator A_f calculates the values of function $f \in M_n$ on the vertices of the shortest chains from $r \in R_n$. These are the chains of length 1 or 2 depending on odd or even values of n . If the values of function f on all elements from set $[l]_n$, $0 \le l \le n$ are known, then according to the monotonity of f , these values are distributed to the set $[l+2]_n$ and according to Hansels's lemma on each chain L of length $l+2$ the values of function f remain unknown for not more than for two vertices. We call these vertices indefinite vertices of L , corresponding to function f . Operator A_f calculates the values of function at these special vertices. This process is applied recursively.

Algorithm F_0 terminates its work, when the values of function *f* are known at all vertices of the chain of length n .

Until now we considered the known means aiming to solve the problem of recovery of monotone Boolean functions $f \in M_n$ with values unknown at all points $\widetilde{\alpha} \in B_n$. Consider a class of monotone Boolean functions of n variables which is narrower than M_n . Suppose that a value k is given, $0 < k < [n/2]$, so that the values of function $f \in M_n$ for vertices of n -dimensional unit cube, which are above the k -th layer equal to 1, and the values are unknown on the reminder area only. Denote this class of function by M_n^k . We consider and adopt the above described technologies in deciphering of this type of monotone Boolean functions.

Theorem. The minimal number $\varphi(n,k)$ of operations of applying to the operator A_f , in recovering the monotone Boolean function's, provided that at all points of n-dimensional cube that are placed higher than some k -th layer, $0 < k < [n/2]$ function equals 1 and on the other points it is unknown, is: $\varphi(n) = C_n^k + C_n^{k-1}$.

Proof. We will show first that $\varphi(n) \ge C_n^k + C_n^{k-1}$ and after that we check $\varphi(n) \le C_n^k + C_n^{k-1}$.

The lower bound will be taken considering some special monotone function:

$$h(\alpha_1, \alpha_2, ..., \alpha_n) = \begin{cases} 1, if\ k \le \sum_{i=1}^{n} \alpha_i \le n \\ 0, if\ 0 \le \sum_{i=1}^{n} \alpha_i \le k-1 \end{cases}$$

We find that $G(h)$, i.e. the deadlock resolving set of function $h(x_1, x_2, ..., x_n)$ contains exactly $C_n^k + C_n^{k-1}$ points as the set of upper zeros are the points of $(k-1)$–th layer of an k-dimensional cube and the set of all lower units will be the points of k-th layer.

We get that $\varphi(n) \geq C_n^k + C_n^{k-1}$.

Now we will prove the upper grade.

It is known that at the points of the cube above the k-th layer function takes the value 1. Consider a chain starting at the k-th layer, we know the values of function at all points on these chains except the starting points, i.e. points located at the k-th layer. By makeing one request for each chain starting from the k-th layer, we will fully recover the function on these chains. Having the values of the function at all points of the chains that start at the k-th layer, we can use the property b) of Hansel's Lemma and property of monotonity to determine the function's values at the points of chains starting at the $k-1$ layer. As a result it turns out that for the chains starting at the $k-1$ layer the functions values remain unknown at no more than two points (see Figure 3).

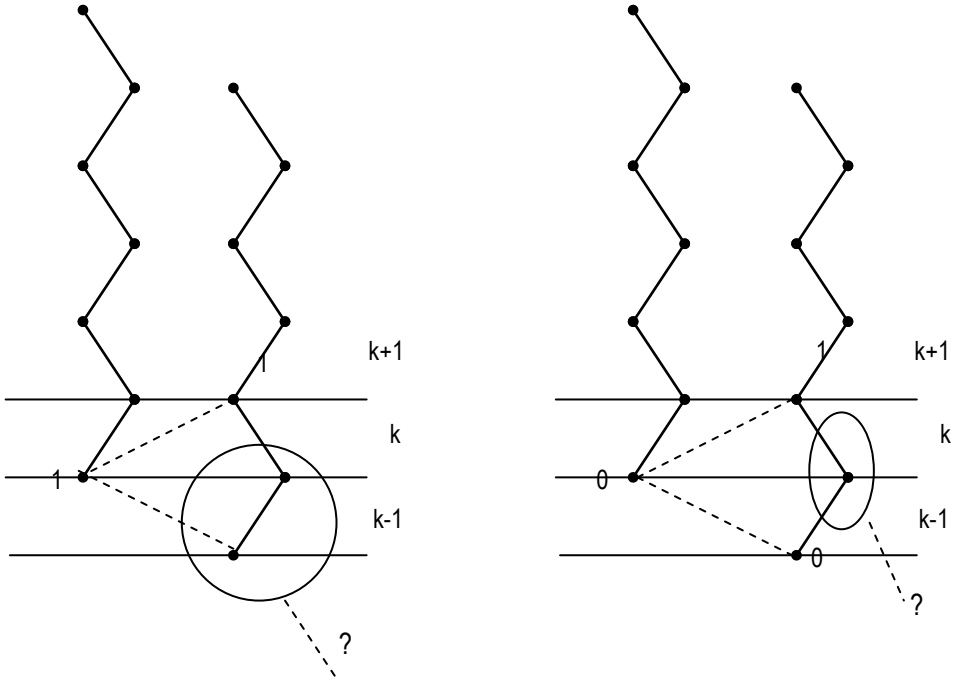

Figure 3.

After performing this procedure for the remaining chains, we conclude that for the complete recognition of a function for the chains starting at the k-th layer is required to determine the function's value only at one point, and that on other chains - at maximum - on two points.

The formula view of the above description is:

φ (n)≤ (the number of chains starting at the k-th layer) +2*(number of other chains starting below the k-th layer).

Let us substitute values using the formula for calculating the number of chains described in the Lemma.

$$\varphi(n) \le C_n^k - C_n^{k-1} + 2(C_n^{k-1} - C_n^{k-2} + ... + C_n^1 - C_n^0 + C_n^0).$$

After making all reductions we receive:

$$\varphi(n) \le C_n^k + C_n^{k-1}.$$

By combining both results we obtain:

$$\left.\begin{array}{l}\varphi(n) \le C_n^k + C_n^{k-1} \\ \varphi(n) \ge C_n^k + C_n^{k-1}\end{array}\right\} \Rightarrow \varphi(n) = C_n^k + C_n^{k-1}.$$

Chain computation

Now we proceed to the collation and analysis of existing knowledge about the chain decomposition of B_n and calculations on chains. Major results here are obtained by G. Tonoyan. These results are connected with the problem of recovering any Boolean function, but they are more universal and applicable to solve other similar problems. Therefore, as a result of our analysis, we will figure out so-called tools i.e. independent procedures that perform particular local operations on the chains, and by the consistent application of which can be solved more global recognition type problems. Specific application will be synthesing of association rules based on the chain split algorithms.

A vertex $\widetilde{\alpha} \in B_n$ is called l-upper zero of function $f \in M_n$, if $f(\widetilde{\alpha}) = 0$ and for any vertex $\widetilde{\beta}$, $\widetilde{\beta} \in \bigcup_{i=2\{n/2\}}^{l} [i]_n = L_l$ from fact that $\widetilde{\beta} \succ \widetilde{\alpha}$ implies that $f(\widetilde{\beta}) = 1$. Here $\{x\}$ denotes the fractional part of x, and $2\{n/2\}$ is to denote the minimal light of chain in chain split, which equals 0 (an isolated point chain) for even n, and 1 for odd n. Remind that $[i]_n$ denotes the set of chains that have length i.

In a similar way a vertex $\widetilde{\alpha} \in B_n$ we call l-lower one of function $f \in M_n$ if $f(\widetilde{\alpha}) = 1$ and for any $\widetilde{\beta} \in L_l$ from the fact that $\widetilde{\beta} \prec \widetilde{\alpha}$ implies that $f(\widetilde{\beta}) = 0$.

We will introduce some definitions that similar but differ somewhat from those already applied.

Let chain $L = (\widetilde{\alpha}_{l+1} \prec ... \prec \widetilde{\alpha}_2 \prec \widetilde{\alpha}_1)$ belongs to R_n. We will say that the length of L is equal to l and $\widetilde{\alpha}_i$ is the i-th vertex of chain L, for $1 \le i \le l+1$.

Let $\widetilde{\alpha} \in L$ and $0 \le k \le \|\widetilde{\alpha}\| - \|\widetilde{\alpha}_{l+1}\|$. Denote by $\widetilde{\alpha}_{(-k)}$ the vertex $\widetilde{\gamma} \in L$ such that $\| \overline{\gamma} \| = \| \overline{\alpha} \| - k$. Through $\widetilde{\alpha}_{(+k)}$ for $\widetilde{\alpha} \in L$ is denoted the vertex $\widetilde{\beta} \in L$ for which $\|\widetilde{\beta}\| = \|\widetilde{\alpha}\| + k$, $0 \le k \le \|\widetilde{\alpha}_1\| - \|\widetilde{\alpha}\|$.

We assume that $\widetilde{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n) \in B_n$ satisfies property Θ if in $(\alpha_1, \alpha_2, ..., \alpha_n)$, for all $k, 1 \le k \le n$, the number of zero coordinates among the first k coordinates are not less than the number of its unit coordinates.

We assume hat $\widetilde{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n) \in B_n$ satisfies property Θ' if $(\overline{\alpha}_1, \overline{\alpha}_2, ..., \overline{\alpha}_n)$ satisfies property Θ, where $\overline{\alpha}_i = \begin{cases} 0, & \alpha_i = 1 \\ 1, & \alpha_i = 0 \end{cases}$.

Vertex, which is obtained from $\widetilde{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n)$ by change of coordinates $\alpha_{i_1}, \alpha_{i_2}, ..., \alpha_{i_s}$ to $\overline{\alpha}_{i_1}, \overline{\alpha}_{i_2}, ..., \overline{\alpha}_{i_s}$ respectively, is denoted as $\widetilde{\alpha}(i_1, i_2, ..., i_s)$.

Finally, a set of numbers $K(\widetilde{\alpha}) = (K_1(\widetilde{\alpha}), K_2(\widetilde{\alpha}), ..., K_n(\widetilde{\alpha}))$ will be associated with vertex $\widetilde{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n) \in B_n$, where:

$$K_1(\overline{\alpha}) = \begin{cases} 2, & \alpha_1 = 0 \\ 1, & \alpha_1 = 1 \end{cases}$$

$$K_i(\overline{\alpha}) = \begin{cases} K_{i-1}(\overline{\alpha}) + 1, & \alpha_i = 0 \\ K_{i-1}(\overline{\alpha}) - 1, & K_{i-1}(\overline{\alpha}) \ge 2,\ \alpha_i = 1 \\ 1 & K_{i-1}(\overline{\alpha}) = \alpha_i = 1 \end{cases}$$

Tool 1 "VERTEX SECUENTIAL NUMBER ON THE CHAIN"

Vertex $\widetilde{\alpha} \in L$ of chain $L \in R_n$ is the $K_n(\widetilde{\alpha})$-th vertex of the chain L.

At the computational level we are given a binary array of memory of length n as input, and use a work numeric array of memory of length n, where numbers saved do not exceed the n. To fill the area by values $K_i(\tilde{\alpha})$ simple comparison operations and $\pm$ are used and the number of this operations is linear by n.

Tool 2 "NEIGHBOUR VERTEX SECUENTIAL NUMBER ON THE CHAIN"

For a vertex $\tilde{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n) \in B_n$ such that $\alpha_{i_0} = 0$ (where $1 \le i_0 \le n$) we have:

1. $K_n(\tilde{\alpha}(i_0))$ *is equal to* $K_n(\tilde{\alpha})$ *if and only if there exist a* j, $j \ge 1$, *such that* $K_{i_0+j}(\tilde{\alpha}) = 1$ *and*
2. $K_n(\tilde{\alpha}(i_0))$ *equals to* $K_n(\tilde{\alpha}) - 1$ *or* $K_n(\tilde{\alpha}) - 2$ *in other cases.*

Tool 3 "LENGTH OF CHAIN ADJASENT TO THE VERTEX NEIGHBOUR TO THE GIVEN ONE"

Vertices adjacent (neighbouring) to vertices of chain L with length l from set R_n belong to chains with lengths $l-2$, l, or $l+2$ from the same set.

Tool 4 "CLIMBING UP"

If there exists a value r, such that $K_r(\tilde{\alpha}) = 1$ and $K_s(\tilde{\alpha}) > 1$ for $r < s < n$, then

$$\tilde{\alpha}_{(+1)} = \begin{cases} \tilde{\alpha}(r+1) & for \quad r < n \\ \phi & for \quad r = n \end{cases}$$

and $\tilde{\alpha}_{(+1)} = \tilde{\alpha}(1)$ if the r mentioned doesn't exist.

Let $H(\tilde{\alpha}) = (n_1, n_2, ..., n_s)$, where $n_1, n_2, ..., n_s$ are all the numbers that satisfy the property $K_{n_i - 1}(\tilde{\alpha}) = K_{n_i}(\tilde{\alpha}) = 1$, $1 \le i \le s, n_1 > n_2 > ... > n_s$, and let $H(\tilde{\alpha}) = \phi$ if such configuration doesn't exist.

Tool 5 "CLIMBING DOWN"

Vertex $\tilde{\alpha}_{(-k)}$ can be determined as follows: if $H(\tilde{\alpha}) = \phi$ then

$$\tilde{\alpha}_{(-k)} = \begin{cases} (0, \alpha_2, ..., \alpha_n) & when\ \alpha_1 = 1\ and\ k = 1 \\ \phi & in\ other\ cases \end{cases},$$

and when $H(\widetilde{\alpha}) = (n_1, n_2, ..., n_s)$, then

$$\widetilde{\alpha}_{(-k)} = \begin{cases} \widetilde{\alpha}(n_k, ..., \alpha_1) & when\ k \le s \\ \widetilde{\alpha}(1, n_s, ..., \alpha_1) & when\ \alpha_1 = 1\ and\ k = s+1 \cdot \\ \phi & in\ other\ cases \end{cases}$$

The set of k-th vertices of all chains with length l from set R_n is denoted by $R(n,l,k)$.

Tool 6 "ALL LOWER VERTICES"

If $R(n,l,l+1) \neq 0,$ then $R(n,l,l+1) = \{\widetilde{\alpha} \in B_n\} / \|\widetilde{\alpha}\| = (n-l)/2\ and\ \widetilde{\alpha}\ satisfies\ property\ \Theta\}.$

Tool 7 "ALL UPPER VERTICES"

$$R(n,l,1) = \{\widetilde{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n) \in B_n\} / (\overline{\alpha}_n, ..., \overline{\alpha}_1) \in R(k+1, l, l+1).$$

Tool 8 "THE CHAIN OF A RELATIVE SUPPLIMENT VERTEX"

Let $\widetilde{\beta} = (\beta_1, \beta_2, ..., \beta_n)$ and $\widetilde{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n)$ are respectively first and k-th, $1 < k \le l+2$, vertices of chain L of length $l+2$ from set R_n. $\widetilde{\alpha}_1 \prec \widetilde{\alpha}_2 \prec \widetilde{\alpha}_3$ is a chain, $\widetilde{\alpha}_1, \widetilde{\alpha}_2 \in L$, $\widetilde{\alpha}'$ is relational supplement of $\widetilde{\alpha}$ regarding to $\widetilde{\alpha}_1$ and $\widetilde{\alpha}_2$, and $H(\widetilde{\beta}) = (n_1, n_2, ..., n_s)$.

Vertex $\widetilde{\alpha}'$ is the $k-1$-th vertex of l length chain, first vertex of which is

$$\widetilde{\gamma} = \begin{cases} \widetilde{\beta}(n_k), & k \le s \\ \widetilde{\beta}(1), & k = s+1 \end{cases}.$$

Consider $\widetilde{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n) \in B_n$ and some set $A \subset B_n$. Introduce the following notations:

$$\vec{\alpha} = \begin{cases} \{\widetilde{\alpha}(k),...,\widetilde{\alpha}(n)\},\ \alpha_k = ... = \alpha_n = 1\ and\ \alpha_{k-1} = 0\ or\ \phi \\ \phi, \quad \alpha_n = 0 \end{cases},$$

$$\overleftarrow{\alpha} = \begin{cases} \{\widetilde{\alpha}(1),...,\widetilde{\alpha}(k)\},\ \alpha_1 = ... = \alpha_k = 0\ and\ \alpha_{k+1} = 1\ or\ \phi \\ \phi, \quad \alpha_1 = 1 \end{cases},$$

$\vec{A} = \{\vec{\alpha} / \widetilde{\alpha} \in A\}$ *and* $\overleftarrow{A} = \{\overleftarrow{\alpha} / \widetilde{\alpha} \in A\}$.

The set of all vertices adjacent to set A is denoted as $(A)_*$ (lower neighbours) and $(A)^*$ (higher neighbours).

Tool 9 "LOWER AND HIGHER ADJACENT VERTICES"

$(R(n,l,l+1))_* = \vec{R}(n,l,l+1)$ and $(R(n,l,1))^* = \overleftarrow{R}(n,l,1)$.

Association rule mining alternatives through the chain split technique

At this point we are given a set of tools that create and use chain split and computations in terms of recognizing of monotone Boolean functions $f \in M_n$ so that values are unknown at all points $\widetilde{\alpha} \in B_n$. We also mentioned that specifically we will consider classes of monotone Boolean functions, narrower than the class M_n of all monotone Boolean functions. Suppose that the values of the function at the points of n-dimensional unit cube, which are above a certain k-th layer ($0 < k < [n/2]$) function takes the value 1, and at other points its value is unknown. We consider our dedicated tools to recognise this type of monotone Boolean functions. We will consider three different options for use of tools.

1. We know that above the layer k function f takes constantly the value 1. Let us consider points of k-th layer. For each such point we will define the length of the chain on which it is situated. We also determine which element of chain it is – its position counting from the start. For this purposes we will use the procedure of calculating $K_n(\widetilde{\alpha})$ for each point of k-th layer. According to Tool 1 vertex $\widetilde{\alpha} \in L, L \in R_n$ is the $K_n(\widetilde{\alpha})$-th vertex of chain L. Then we can determine the length of chain as follows. It is known that the point $\widetilde{\alpha}$ has k ones and $n-k$ zeros, where $0 \le k \le n$ and it is known that $\overline{\alpha}$ is the i-th vertex of its chain. It is easy to calculate that the first (biggest, upper) vertex of the chain must contain $(k+i-1)$ ones, and accordingly $n-(k+i-1)$ zeros. Taking into account the symmetry of chains relative to the middle layers, we can calculate that the last vertex of the chain consists of $n-(k+i-1)$ ones and $(k+i-1)$ zeros. Since the chain is obtained by changing one zero in the coordinates of the vertices to one at each crossing of a layer, the length of

chain will be equal to the difference between the number of ones in the first and last vertices of the chain, i.e. for the chain $L \in R_n$ we find that its length is equal to $l = n - 2(k + i - 1)$.

From all points of the k-th layer we mark those that are the last vertices of their chains. For these vertices the length of the chains that pass these vertices is $n - k - k = n - 2k$. With the help of the operator A_f the values of function at these points are determined. Next, consider chains of length $l + 2$ and their lower parts that are below the layer k. According to the symmetry of chains we have that chain of length $l + 2$ end at $(k - 1)$-th layer. Let us consider all points of the *(k-1)*-th layer. Using Tool 6 we will construct the set $R(n, l + 2, l + 3)$ that is the set of all last points of chains with length $l + 2$. $R(n, l + 2, l + 3) = \{\widetilde{\alpha} \in B_n / \|\widetilde{\alpha}\| = (n - l - 2)/2\}$ and $\widetilde{\alpha}$ satisfies the property $\Theta\}$. For each vertex $\widetilde{\alpha} \in R(n, l + 2, l + 3)$ after applying $\widetilde{\alpha}_{(+1)}$ $l + 2$ times we find the appropriate first vertex $\widetilde{\beta}$ of chain that contains $\widetilde{\alpha}$ with and have length $l + 2$. We also find the $l + 2$-nd vertex of considered chain, i.e. the point $\widetilde{\delta}$ that is preceds $\widetilde{\alpha}$. This point is situated at k-th layer and for its determination it is sufficient to apply $\widetilde{\alpha}_{(+1)}$ once. Now let us turn to Tool 8. Let us $\widetilde{\beta} = (\widetilde{\beta}_1, \widetilde{\beta}_2, ..., \widetilde{\beta}_n)$ and $\widetilde{\delta} = (\widetilde{\delta}_1, \widetilde{\delta}_2, ..., \widetilde{\delta}_n)$ are respectively the first and the k-th, $1 < k \le l + 2$, vertices of chain L of length $l + 2$ that belongs to R_n. $\widetilde{\alpha}_1 \prec \widetilde{\delta} \prec \widetilde{\alpha}_2$ is a chain and $\widetilde{\alpha}_1, \widetilde{\alpha}_2 \in L$, $\widetilde{\delta}'$ is the relative supplement of $\widetilde{\delta}$ in regard to $\widetilde{\alpha}_1$ and $\widetilde{\alpha}_2$, and $H(\widetilde{\beta}) = (n_1, ..., n_s)$ is a vector as defined in Tool 5. In this conditions $\widetilde{\delta}'$ is $(k - 1)$-th vertex of a chain with length l, the first vertex of which is

$$\widetilde{\gamma} = \begin{cases} \widetilde{\beta}(n_k), & k \le s \\ \widetilde{\beta}(1), & k = s + 1 \end{cases}$$

For each matched point $\widetilde{\beta}$, that is the point that precedes $\widetilde{\alpha}$, we find the corresponding point $\widetilde{\gamma}$ as described above. From each $\widetilde{\gamma}$ according to the rule $\widetilde{\alpha}_{(-k)}$ we define the point $\widetilde{\delta}'$, which is a supplement to point $\widetilde{\delta}$. Point $\widetilde{\delta}$ is the last vertex of a chain with length l and is placed on the k-th layer. The value of function at this point has already been calculated by the operator A_f. We extend this value by the property of monotonity to the points of chain with length $l + 2$. Remaining unknown points of chain with length $l + 2$ will be defined using the operator A_f.

At the general step, we consider a chain of length $l + m$ and find its first and last vertices, then is finding the corresponding first vertex of chain with length $l + m - 2$, and its last point, which is supplement for the penultimate point of a chain with length $l + m$. At this stage the values of function at all points of the chain with

length $l+m-2$ are already known, and it remains only to extend these values to the chains with length $l+m$ and using the operator A_f to compute the value of function on the remaining unknown points.

2. As in the previous paragraph we will start the function recognition from the k-th layer. We define the points of k-th layer, which are the last points for chains of length l and applying to A_f we calculate the values of the function at these points. We now turn to the chains of length $l+2$, and build the set $R(n,l+2,l+3)$. According to Tool 7:

$$R(n,l+2,1)=\{\widetilde{\alpha}=(\alpha_1,\alpha_2,...,\alpha_n)\in B_n\ /(\overline{\alpha}_n,\overline{\alpha}_{n-1},...,\overline{\alpha}_1,)\in R(n,l+2,l+3)\}.$$

By apply $\widetilde{\alpha}_{(-(l+2))}$ to each vertex $\widetilde{\alpha}\in R(n,l+2,1)$ finding $(l+2)$- nd vertex of chain with length $(l+2)$, which is the relational supplement for $(l+1)$-st vertex of chain with length l, the first vertex of which is obtained by the Tool 8.This method of finding the first vertex makes the task easier compared with the method described in paragraph 1, because of there, for determining the first vertex was necessary to pass step by step through the whole chain starting from the last vertex using $\widetilde{\alpha}_{(+1)}$. In general these methods are identical.

3. Consider another way of recognition, which differs from the first two in the initial notation of the problem. Consider an n-dimensional unit cube B_n, each vertex $\widetilde{\alpha}=(\alpha_1,\alpha_2,...,\alpha_{n)}$ of which is composed of ones and zeros, where 0 indicates that the α_i-th element is involved in the transaction, and 1 - no. Value of the function on the vertex $\widetilde{\alpha}=(\alpha_1,\alpha_2,...,\alpha_{n)}$ is equal to 1 if the set is frequent and 0 otherwise. In these definitions the frequent sets move to the top of the cube, and the function to be recognized takes the following form: on points of B_n that are placed below a certain k-th layer, where $n/2<k\le n$ function takes value 0, and on the rest of cube its values are unknown. All other assumptions and definition of the problem remain unchanged. The recognition process starts from chains that begin at the k-th layer. All points of k-th layer are considered, and for these points corresponding values $K_n(\widetilde{\alpha})$ are calculated. Those points are selected for which $K_n(\widetilde{\alpha})=1$, i.e. points, which are the first points for the chains starting at the k-th layer. Then the lengths of these chains are calculated and the value of function at these points are determined by applying to the operator A_f. We assume the length of chains we consider is l. After that, we consider chains of length $l+2$, i.e., starting at the $(k+1)$-th layer. For this we consider all points of the $k+1$- th layer and select those who have $K_n(\overline{\alpha})$ = 1. The second points of chains with length $l+2$ are found using operation $\widetilde{\alpha}_{(-1)}$. Relational supplements of these points are situated on chains of length l, and are the first points of these chains, and the function values of which we already know. Relative supplements are determined by Tool 8. We extend the function values by monotonity

to the points of chains with length $l+2$. Values of the function at the points remaining unknown are calculated with the help of the operator A_f. We continue the consideration of chains until no unknown points remain.

Software Implementation

For the practical implementation of the above-mentioned methods in systems of association rules mining, a software application was initiated. For this reason an open programming environment (open source) was identified, which is dedicated to the tasks of data analysis, known as "Orange Canvas". This system includes algorithms of commonly used Data Mining techniques i.e. classification, clustering, decision trees, association rules searching, etc. As an algorithm for association rules mining it includes some modification of the Apriori algorithm.

As noted above, we have proposed an alternative approach for solving the problem of association rules mining based on monotone Boolean functions, n-dimensional cube and its coverage with Hansel's chains. Without using the results achieved by Tonoyan it is mostly impossible to make effective program implementation of the proposed method, since for large n operational resources of regular computers are not sufficient to store the current computational data. The reason for this is that for Hansel's technique requires to build the cube completely, as well as to build its chain coverage and keeping it in memory. The application of Tonoyan's Tools allows instead of permanent storage of a cube and its coverage, to calculate the required points of the cube i.e. the first and last vertices of the chains, adjacent vertices of a given point, relative supplement of a given point at a certain interval, etc.

For complete data processing, sometimes it is not enough to apply the association rules technique. It was therefore decided not only to create a software implementation of the algorithm, but to introduce it into the existing Orange Canvas, where a number of data analysis approaches are implemented already.

Bibliography

[KOR,1965] B. K. Korobkov. On monotone functions of algebra of logic, Problemy Kibernetiki, Nauka, Moscow, v. 13, pp. 5-28 (1965).

[AS, 1979] L. Aslanyan. Isoperimetry problem and related extremal problems of discrete spaces, Problemy Kibernetiki, Nauka, Moscow, v. 36, pp. 85-126 (1976).

[HANS, 1966] G. Hansel. Sur le nombre des functions booleennes monotones de n variables, C.R. Acad. Sci. Paris, 262, serie A (1966), pp. 1080-1090.

[TON, 1976] G. P. Tonoyan. Chain decomposition of n dimensional unit cube and reconstruction of monotone Boolean functions, JVM&F, v. 19, No. 6, 1532-1542 (1976).

[AGRAW, 1996] R. Agrawal, H. Mannila, R. Srikant, H. Toivone and A. I. Verkamo. Fast discovery of association rules. In U. M. Fayyed, G. Piatetsky-Sharpiro, P. Smyth, and R. Uthurusamy, editors, Advances in knowledge Discovery and Data Mining, AAAI/MIT press, pp. 307-328, 1996.

[KOT, 2006] S. Kotsiantis, D. Kanellopoulos. Association Rules Mining: A Recent Overview, GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 71-82.

[AS, 2008] L. Aslanyan and R. Khachatryan. Association rule mining inforced by the chain decomposition of an n-cube, Mathematical Problems of Computer Science, XXX, 2008, ISSN 0131-4645.

[KACH, 2008] R. Khachatryan. Data Mining algorithmic alternatives, in "Proceedings of the Young Scientists Conference on Perspectives for development of molecular and cellular biology" , ISBN 978-5-8080-0724-6, pp. 54-55, Yerevan, 2008.

[ASL-SAH, 2009) L. Aslanyan, H. Sahakyan, Chain split and computations in practical rule mining, International Book Series "Information Science and Computing" , Book 8, "Classification, Forecasting, Data Mining", ISSN: 1313-0455, 132-135(2009).

[ASL-SAH, 2010] L. Aslanyan, H. Sahakyan. On structural recognition with logic and discrete analysis, NIT 2010, Madrid Technical University, 28 Sept. – 2 Oct. 2010, International Journal "Information Theories & Applications", volume 17, number 1, 3-9(2010).

[SAH-ASL, 2010] H. Sahakyan, L. Aslanyan. Chain Split of Partial Ordered Set of k-Subsets, International Book Series "Information Science and Computing" , Book 18, "New Trends in Information Technologies", ITHEA, ISSN: 1313-0455, 55-65(2010).

Authors' Information

Levon Aslanyan *– Head of Department, Institute for Informatics and Automation Problems, P.Sevak St. 1, Yerevan 14, Armenia, e-mail: lasl@.sci.am*

Robert Khachatryan *– Researcher, Institute for Informatics and Automation Problems, P.Sevak St. 1, Yerevan 14, Armenia, e-mail: robert_khachatryan@hotmail.com*