CUBE-SPLIT TECHNIQUE IN QUANTITATIVE ASSOCIATION RULE MINING

Levon Aslanyan, Hasmik Sahakyan

***Abstract**: In this paper we consider association rules mining in tables with quantitative attributes. The chain split technique of finding frequent itemsets, known in Boolean association rule mining domain, is extended to the "cube-split" technique being used for finding frequent itemsets for the case of specific quantitative attributes.*

***Keywords**: Data mining, frequent itemsets, quantitative association rules, chain-split technique, cube-split technique.*

***ITHEA Keywords**: I.5 Pattern recognition: I.5.1 Models*

1. Introduction

The goal of the data mining and knowledge discovery is to extract high level information relationships in raw big data of applications. Logical-Statistical Association rules (simply association rules) are one of the key data mining techniques. An association rule is a logical-statistical relation of the form $A \Rightarrow B$, where A and B are events. The rule states that with a certain probability, called the ***confidence*** of the rule, when A occurs in the given database so does B. The second important characterization of rules is the probability of occurrence of the event $A\&B$ over the database records, called the rule ***support***. The problem is to find all association rules that satisfy the requirement of minimum support and minimum confidence. The main bottlenecks of the rule generation are the speed of the algorithms used, the limitation on subset of the rule collections obtained, and the applied value of these rules – their interpretation. A well-known interpretation of association rules comes with the problem market-basket-data-analysis.

The structure of this article is as follows: Boolean association rule mining problem is considered in Section 2, and a brief description of the APRIORI algorithm, the de facto standard approach for the case of binary attributes, - is introduced. Section 3 describes the necessary structural knowledge on the discrete grids that is necessary for the functionality of the chain split technique which is the main instrument considered in this work. Section 4 is devoted to the analysis of rule mining with quantitative attributes. The chain split technique is extended to the case of quantitative attribute rule mining.

2. Boolean Association Rule Mining

About the problem

Association rule mining in terms of confidence/support was first introduced by [Agrawal et al., 1993], and later is addressed in [Agrawal et al., 1994]. These first papers consider databases consisting of categorical attributes only (a *categorical attribute* is one which contains discrete, and typically, unordered data). Thus the events A and B on both sides of the rule $A \Rightarrow B$, are logical expressions of categorical variables. *The aim is to find all rules with confidence and support above the user-defined thresholds of these parameters* (**minconf** and **minsup**). It is desirable that the set of constructed rules is sharply limited in size. Then the rules may have a better interpretation.

Earlier, for the purpose of discovering exact or almost exact rules Piatetsky-Shapiro [Piatetsky-Shapiro, 1991] introduced three principles for rule interestingness (0 if the variables are statistically independent, monotonically increasing if the variables occur more often together, and monotonically decreasing if one of the variables alone occurs more often) defining in this way the base concepts of contemporary rule mining. In structural data analysis rules are likely to be exact. Exact rules appear also as functional dependencies in relational databases [Armstrong, 1998]. On the other hand, in business databases and in very large databases (similar to supermarket transaction databases), rules may be descriptive even in approximation, with the confidence much less than 100%. A number of efficient algorithms for mining binary association rules have been developed (see [Agrawal et al., 1994], [Mannila et al., 1994], [Toivonen, 1996] for just a few examples). And the APRIORI [Agrawal et al., 1996] algorithm is known as the de facto standard of Boolean association rule mining.

Description

Consider a set $I = \{x_1, x_2, \dots, x_n\}$ consisting of n items x_i, and their subsets (itemsets) $X \subseteq I$. We say that it is given a k-itemset, when $|X| = k$. Let D be a database of records (transactions) that are itemsets.

We say that the record $T \in D$ is contributing to the itemset X, if $X \subseteq T$. Association rule is an "if-then" type logical rule $X \Longrightarrow Y$, the fulfillment of which is related to the certain (statistical) conditions. Let X and Y be itemsets where $X \cap Y = 0$. The ratio of the number of all records of D contributing to X and the overall number of records of D - is called support of X in D:

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

Next to this is the concept of support for the rule $X \Longrightarrow Y$ itself:

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

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

$$\mathrm{conf}(X \Longrightarrow Y) = \sup(X \cup Y)\,/\sup\,(X),$$

which is the conditional probability that a record contains Y when it is known that it contains X.

APRIORI algorithm

Practical implementation of association rule mining techniques is a subject of intensive theoretical and algorithmic studies. It is well known that the problem splits naturally into two stages [Agrawal et al., 1996]. The first stage is constructing the so called **frequent fragments** (itemsets), 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** with a given confidence from the set of frequent subsets constructed during the first stage of the algorithm.

The well recognized algorithm in association rules mining APRIORI [Agrawal et al., 1996] builds the set of frequent subsets (first stage mentioned above) with so-called *building up* method. APRIORI first considers one-element subsets, and computes their frequencies. Next to this, it considers all two-element subsets one-element subsets of which are frequent, and verifies their proper occurrences in the database. Thus the frequent subsets can be building up to the state when it includes subsets that all are not frequent enough. This procedure is known as ***growing*** of frequent itemsets. Computational complexity here is significant and it is especially important because of algorithms will be used over the very large data volumes.

Alternative approaches

Are there any alternative approaches for building rules? There is a huge number of approaches, ideas and algorithms that address this issue. The case of quantitative attributes is much harder. General approaches applied are: rule interpretation as a union of parts – “population-subset” and “extraordinary-behavior” with consecutive optimization (maximization of support and/or confidence); design and consideration of interpretable regions such as attribute and multi-attribute “convex” regions; and - application of fast algorithms, for example randomized algorithms, OPUS, Kadane's, Elias’s and other algorithms.

In this paper we propose a new approach to rule mining which connects with the well-known results from the geometry of the n-dimensional unit cube that corresponds to the Hansel’s algorithm for monotone Boolean function identification. We give an extension of this approach to the multidimensional multivalued grid [Aslanyan et.al., 2017]. The extension will cover the case of rule mining by the sets of numerical attributes.

Monotonicity property

Monotonicity is the important property of function $\mathrm{sup}(\mathrm{X})$,

$$(\mathrm{X} \subseteq \mathrm{Y}) \Rightarrow \mathrm{sup}(\mathrm{X}) \geq \mathrm{sup}(\mathrm{Y}).$$

Monotonicity is the key property of the Boolean rule mining models, but it is not simply interpretable in the case of numerical attributes. Things changed when we try to speak in terms of attribute negations [Boulicaut, 2001]. This is when we aim at knowing the frequencies of conjunctions of all literals – that means attributes and their negations. In computational layer this doubles the number of attributes and forces the flooding of area of frequent subsets. [Boulicaut, 2001] and other publications derived formulas that introduce frequencies for negations in terms of formulas of positive frequencies. The issue of positive-negative attributes appears in case of quantitative attributes as the monotone-anti-monotone behavior. Quantitative-monotone, in its turn, means that the increase of attribute values increases probability of the target event. For a particular target event we suppose that all individual attributes are monotonic and that they, in integration, also monotonically depend on the target event.

Isoareas

Isoareas of $\mathrm{sup}(\mathrm{X})$ in Figure 1 can be presented by the sequences of embedded monotone Boolean functions. But such sequences cannot behave arbitrarily. For example, if we consider a simple 2-row database then these rows correspond to 2 vertices of n-cube, let they be denoted by v_1 and v_2. Now consider the subcubes $I_1(v_1, \tilde{1})$ and $I_1(v_2, \tilde{1})$ formed by these vertices and by the $\tilde{1} = (1,1,\cdots,1)$ vertex of the cube. All points of these subcubes 1-support one of the rows but the points in intersection support both of them. So it is not possible to design a database that gives homogeneous support value over the area covered by $I_1(v_1, \tilde{1})$ and $I_1(v_2, \tilde{1})$. So the mentioned “sequences of embedded monotone Boolean functions” are a specific inclusion-exclusion type objects to be described and studied in deep.

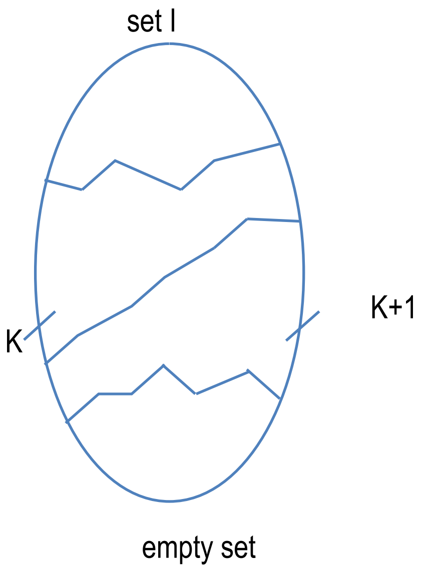

Figure 1. Hasse type diagram of power set of set I of attributes with isoareas of the values of function $\sup(\mathrm{X})$

Frequent Subsets Mining **(FSSM) Problem**

Given: $\mathcal{D}$ – the database of subsets of some set A of attributes; and a frequency threshold $t \geq 0$.
Provide: the t-frequent subsets in $\mathcal{D}$.
For the FSSM Problem several approaches are proposed: variations to the original APRIORI algorithm [Agrawal et al., 1996], as well as other approaches like [Aumann et al., 2003]. The most efficient algorithms are based on the observation that the frequent subsets are determined by the closed frequent subsets, and such algorithms need to mine the closed frequent subsets at first [Li et al., 2006].

A subset $\mathrm{X} \subseteq \mathrm{A}$ is called *closed* (maximal subset) if for each $\mathrm{Y} \subseteq \mathrm{A}$, $\mathrm{Y} \supset \mathrm{X}$, it holds $\sup(\mathrm{Y}) < sup(\mathrm{X})$.

For $\mathrm{X} \subseteq \mathrm{A}$ we define $\rho(\mathrm{X}) \coloneqq \bigcap\{\mathrm{Y} \subseteq \mathrm{A} : \mathrm{Y} \supseteq \mathrm{X} \text{ and } \mathcal{D}(\mathrm{Y}) > 0\}$ to be the closure of X in $\mathcal{D}$. It can be checked that $\mathrm{X} \subseteq \rho(\mathrm{X}), \mathrm{X} \subseteq \mathrm{Y} \Rightarrow \rho(\mathrm{X}) \subseteq \rho(\mathrm{Y})$ and $\rho\big(\rho(\mathrm{X})\big) = \rho(\mathrm{X})$. It also can be checked that the closed subsets are the closures.

Obviously $\mathcal{D}: 2^{\mathrm{A}} \to \mathbb{N}_0$ defines a monotonic decreasing integer function when X increases by the set-inclusion, over the Boolean cube $(2^{\mathrm{A}}, \subseteq)$, and it can be checked that the frequent subsets correspond to the subsets of the closed frequent subsets, and thereby closed frequent subsets determine the frequent subsets.

3. On the Geometry of the n-Dimensional Unit Cube

n dimensional unite 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 the n-th Cartesian degree of the set B. B^n is the set of all binary vectors $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$, which are called *vertices* or points. Usually B^n is presented geometrically via the Hasse diagram, in which vertices of B^n are placed in horizontal layers; the k-th layer $(0 \le k \le n)$ contains all the vertices with k number of ones. The layers are arranged vertically starting from the zero layer (at the bottom) to the layer with number . The k-th layer consists of C_n^k vertices. Two vertices $\tilde{\alpha}$ and $\tilde{\beta}$ are called adjacent if they differ in exactly one coordinate. These neighboring vertices are connected by an edge.

Vertices of B^n are organized as follows: a point $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ of B^n precedes the point $\tilde{\beta} = (\beta_1, \beta_2, \dots, \beta_n)$ of B^n, if $\alpha_i \le \beta_i, 1 \le i \le n$. The fact that a point $\tilde{\alpha}$ precedes the point $\tilde{\beta}$ is denoted by $\tilde{\alpha} \preccurlyeq \tilde{\beta}$. If $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ and $\tilde{\alpha} \ne \tilde{\beta}$ then we write $\tilde{\alpha} \prec \tilde{\beta}$. Two different points $\tilde{\alpha}$ and $\tilde{\beta}$ are called *comparable* if one of the following conditions occur: $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ or $\tilde{\beta} \preccurlyeq \tilde{\alpha}$. Otherwise they are *incomparable*.

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. But 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 very small subset of vertices 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 all layers of B^n.

Monotone Boolean functions

Boolean function $f(x_1, x_2, \dots, x_n)$ is called *monotone* if from the fact that $\tilde{\alpha} \prec \tilde{\beta}$ it implies that $f(\alpha_1, \alpha_2, \dots, \alpha_n) \le f(\beta_1, \beta, \dots, \beta_n)$. 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 the vertices of $\hat{f}^0$. Geometrically the area of 0 assignments of f is a union of *subcubes*, composed by the vertex $\tilde{0}$ and the vertices of $\hat{f}^0$. Another important property is that on growing chains of vertices in B^n, the function values - 0's and 1's fills two different intervals at most.

Two type of *recognition problems* about the monotone Boolean functions are rising in different applications. One is the recognition whether the given $f(\tilde{x})$ ***belongs to*** $\boldsymbol{M_n}$, the class of all monotone Boolean functions; and the second is in ***deciphering of*** $\boldsymbol{f(\tilde{x})}$ itself given that $f(\tilde{x}) \in M_n$. We address the second topic because of its identity to the problem of frequent itemset mining.

Chain Split

A separate group of algorithms for Boolean association rule mining is introduced in [Aslanyan et al., 2008].

Suppose that an arbitrary (unknown to us) function $f(\tilde{x}) \in M_n$ is given by an operator A_f, which returns 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 $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n) \in B^n$ other points of B^n become determined through the extension by the monotonicity 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 $f(x_1, x_2, \dots, x_n)$ an algorithm from R exploiting the operator A_f restores the complete table of values of $f(\tilde{x})$. Obviously the work of algorithms consists of several stages. Algorithm selects a point $\tilde{\alpha} \in B^n$ and with help of operator A_f computes the value $f(\alpha_1, \alpha_2, \dots, \alpha_n)$ (*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 monotonicity, 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 $\tilde{\beta}$ that are higher that $\tilde{\alpha}$ (according to the order of vertices defined above) $f(\tilde{\beta}) = 1$ 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 monotonicity. This process is repeated until the table of values is filled completely.

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

dimensional monotone functions can be characterized by the 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

a) the function g belongs to N,

b) 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 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 $\tilde{\beta}$ satisfying the condition $\tilde{\alpha} \prec \tilde{\beta}$, and by $L(\tilde{\alpha})$ - the set of points $\tilde{\gamma}$ such that $\tilde{\gamma} \prec \tilde{\alpha}$.

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

The lower one of a monotone function $f(x_1, x_2, \dots, x_n)$ 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 $f(x_1, x_2, \dots, x_n)$, 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, for instance in class of symmetric Boolean functions that we mentioned above). This deadlock resolving set for a monotone Boolean function is the set $G(f) = Z(f) \cup O(f)$.

A brief characterization of the chain split approach is as follows. The n-dimensional unit cube B^n is a binary 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 [Hansel, 1966] 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 [Tonoyan, 1976] offered a computational approach to the work with chains. This is fundamentally and significantly simplifying the recognition algorithm although the 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 the chain split.

The global 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 and extend this technique to the mining problems of numerical attributes. 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 knowing 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.

4. Quantitative Association Rule Mining

In practice, many, if not most, databases contain quantitative data. ***Unfortunately, the definition of categorical association rules does not translate directly to the quantitative case.*** This initiates intensive search for a definition or model of association rules for the case of databases with quantitative attributes. [Srikant et al., 1996] made an approach to extend the categorical rule definition to include quantitative data. They used some kind of grouping and discretization in intervals of values of attributes. Thus, each basic event becomes either a categorical item or a range of numerical values. This way, although the base approach is powerful, raises a number of drawbacks in form of correct interval composition or as the exponential blowup of the number of the rules generated.

[Fukuda et al., 1996a] considered a different perspective to quantitative association rule mining problem, providing efficient algorithms by the given values of ***minconf*** and ***minsup.*** They use computational geometry methods achieving efficiency even for very large size databases, but the rules considered are plain: with one categorical attribute at the right side. The left side event of the rule have the form $A \in [v_1, v_2]$ with an attribute A and its values v_1, v_2. The technique applied use randomized construction of intervals, combining them into the equi-depth buckets, then joining consecutive buckets into the regions of high support. The main target is to provide the required minconf, maximize the region support, and to keep the algorithms at linear complexity.

Next approach to the problem of quantitative association rules is derived by [Aumann et al., 2003]. The idea used is to compute and apply databases statistical values to increase rule interestingness and to combat the flooding of the number of generated rules. This paper well summarizes several data mining concepts. First of all, it brings the following useful description: An association rule indicates association between ***a subset of the population*** described by the left-hand side of the rule, and an ***extraordinary behavior of this subset*** described by the right-hand side of rule. [Webb, 2001] later ([Aumann et al., 2003] appeared as a conference publication at 1999) labelled these rules as impact rules to outline the difference to the approach [Srikant et al., 1996].

Thus, the introduced general structure of an association rule is of form:

population-subset ⇒extraordinary behavior. ***(1)***

In summary, an association rule considered is a rule of the form: "***population-subset"⇒"mean of values for the subset",*** where population-subset is large enough and the mean of the subset is significantly different to the mean of its complement in the database to form an extraordinary event. And we indeed foresee the efficient algorithms that are able to generate the required rules of this type. The general structure (1) gives rise to many different concrete rule types, determined by the subset class used on the left-hand side, and the description of extraordinary used for the right-hand side. Continuing in this way [Aumann et al., 2003] constructed two types of rules: Categorical to Quantitative rules with an unlimited number of attributes on each side, and Quantitative to Quantitative where both sides contain a single attribute only. It is evident that still these are narrow rule sets. Information on other work on quantitative rule mining may be found at [Aumann et al., 2003] and [Hahsler, 2017]. In particular, [Fukuda et al., 1996b] and [Yoda et al., 1997] consider rules, from 2 numerical to one Boolean attributes, where numerical attributes construct a connected and x-monotone, rectangular, or rectilinear areas (in terms of discrete tomography x-monotone means v-convex, and rectilinear means hv-convex). The whole diversity of studies, as we see, consider a large number of very restricted types of rules, and models, that rarely provide effective computation of the limited number of properly optimized rules. Another concern is that these quantitative approaches loose the main essence of the Boolean rule mining as is the frequent itemset growing.

Let us mention also QuantMiner [Salleb-Aouissi et al., 2007] that is a Quantitative Association Rules Mining tool available online. It takes into consideration a set of numerical attributes in the mining process without a prior binning/discretization of the data. It exploits a recent and innovative research in genetic algorithms.

Concluding,

(*) In this context we aim at constructing ***(multi&numerical)*** ⟹ ***Binary*** rules under the unique supposition of attribute monotonicity. Monotonicity is a natural phenomenon not absolute but it can be supposed for any kind of extraordinary behavior. In exceptional cases the attribute value domain can be split in several intervals that will support monotonicity. All we suppose is that these cases of split for monotonicity are only exceptions and that the individual attribute monotonicity integrates in a collective monotonicity of the model. As a consequence, we will receive a model based on frequent itemset growing technique. And we will extend the Hansel's chain split technique and the monotone Boolean function recognition as the algorithmic basis of our newly quantitative association rule mining model.

Multi-valued cube splitting to unit-cubes

Let $\widetilde{m} = (m_1, m_2, \dots, m_n)$ be an integer vector of n dimensions, and $\Xi_{\widetilde{m}}^n$ be the set of vertices of the n coordinate discrete grid defined as the Cartesian product of sets $\Xi_{m_i} = \{0,1, \dots, m_i - 1\}$:

$$\Xi_{\widetilde{m}}^n = \Xi_{m_1} \times \Xi_{m_2} \times \dots \times \Xi_{m_n} = \{(a_1, a_2, \dots, a_n) : a_i \in \Xi_{m_i}, i \in \overline{1,n}\}.$$

In this section we introduce a special decomposition of $\Xi_{\widetilde{m}}^n$ into the structures isomorphic to binary cubes.

Binary cubes may have different dimensions but their distribution by the cube-size is canonical and the cubes in total cover the $\Xi_{\widetilde{m}}^n$ disjointly and entirely. Before descriptions we distinguish several type of special vertices in $\Xi_{\widetilde{m}}^n$.

Middle vertices

Vertices $\widetilde{m}_{mid+} = (\lceil \frac{m_1}{2} \rceil, \lceil \frac{m_2}{2} \rceil, \dots, \lceil \frac{m_n}{2} \rceil)$ and $\widetilde{m}_{mid-} = (\lfloor \frac{m_1}{2} \rfloor, \lfloor \frac{m_2}{2} \rfloor, \dots, \lfloor \frac{m_n}{2} \rfloor)$ we call middle vertices of $\Xi_{\widetilde{m}}^n$. These two vectors coincide when all m_i are even values. Being skewed because of the possible differences of values $m_1, m_2, \dots, m_n$, $\Xi^n_{\widetilde{m}}$ has exactly the mentioned one or two special points at the center. Even with one odd m_i the central points are different. Let $n_{\neq}$ denote the number of all odd m_i values. The set of all points with coordinates $m_{imid+} = \left\lceil \frac{m_i}{2} \right\rceil$ and/or $m_{imid-} = \left\lfloor \frac{m_i}{2} \right\rfloor$ are allocated between the $\widetilde{m}_{mid+}$ and $\widetilde{m}_{mid-}$. Number of such vertices is equal to $2^{n_{\neq}}$. These points fill a structure isomorphic to $B^{n_{\neq}}$. Symmetrically, let us denote $n - n_{\neq}$ by $n_{=}$.

Upper vertices, lower vertices

A vertex $(a_1, a_2, \dots, a_n)$ of $\Xi^n_{\widetilde{m}}$ is called upper vertex if $(a_1, a_2, \dots, a_n) \geq \widetilde{m}_{mid+}$. Similarly, vertex $(a_1, a_2, \dots, a_n)$ of $\Xi^n_{\widetilde{m}}$ is called lower vertex if $(a_1, a_2, \dots, a_n) \leq \widetilde{m}_{mid-}$. $\hat{\Xi}$ and $\check{\Xi}$ denote the sets of all upper and lower vertices of $\Xi^n_{\widetilde{m}}$, correspondingly. It is easy to check, that

$$\left|\hat{\Xi}\right| = \left|\check{\Xi}\right| = \prod_{i=1}^{n}(\lfloor m_i/2 \rfloor + 1).$$

Vertical equivalence

Vertices $\tilde{a} = (a_1, a_2, \dots, a_n)$ and $\tilde{b} = (b_1, b_2, \dots, b_n)$ of $\Xi^n_{\widetilde{m}}$ are called vertically equivalent if $a_i \in \{b_i, m_i - b_i\}$ for $1 \leq i \leq n$. It is easy to check that this condition is to symmetrically apply to $\tilde{a}$ and $\tilde{b}$, creating a structure of equivalence classes over the $\Xi^n_{\widetilde{m}}$. Let $V(\tilde{a})$ denote the class of V-equivalence of vertex $\tilde{a}$. In $V(\tilde{a})$ we distinguish two vertices $\hat{a}$ and $\check{a}$ with coordinates defined as follows:

$$\hat{a}_i = \begin{cases} a_i & if\ a_i \geq m_{imid+} \\ m_i - a_i & if\ a_i \leq m_{imid-} \end{cases}$$

$$\check{a}_i = \begin{cases} m_i - a_i & if\ a_i \geq m_{imid+} \\ a_i & if\ a_i \leq m_{imid-} \end{cases}$$

Vertices $\hat{a}$ and $\check{a}$ are the only "two" vertices for an arbitrary $V(\tilde{a})$ that belong to $\hat{\Xi}$ and $\check{\Xi}$, respectively. Thus all vertices of sets $V(\tilde{a})$ can be extended from the upper and/or lower elements of the class of V-equivalency by component subset inversions (in respect to values m_i). It is evident that the equivalence classes of different vertices of $\hat{\Xi}$ (or $\check{\Xi}$) are disjoint. This construction provides partitioning of $\Xi^n_{\widetilde{m}}$ into $\left|\hat{\Xi}\right|$ equivalence classes uniquely defined by the elements of $\hat{\Xi}$.

For a given $\tilde{a} \in \Xi_{\tilde{m}}^n$ define an integer $k_{\neq} = |\{a_i : a_i \neq m_i/2\}|$. Then $|V(\tilde{a})| = 2^{k_{\neq}}$. We identify each vertex $\tilde{\beta} \in V(\tilde{a})$ with an n-dimensional binary sequence γ, such that $\tilde{\gamma}_i = 1$ if and only if $\beta_i = \hat{\alpha}_i$. In this manner, $V(\tilde{a})$ becomes isomorphic to the $k_{\neq}$-dimensional binary cube $B^{k_{\neq}}$: the 0-th level contains the lower vertex of $V(\tilde{a})$ belonging to $\breve{\Xi}$; the i-th level consists of all vertices of $V(\tilde{a})$ which can be obtained from the lower vertex by applying i number of component inversions.

Thus $\Xi_{\tilde{m}}^n$ is partitioned into $|\hat{\Xi}|$ disjoint equivalence classes - that are identical in structure to the binary cubes. It is worth to mention that in usual chain-split (as is the partition of the binary cube in [Hansel, 1966]) vertices in chains are composed of neighbor vertices, whereas in the case of cube-split edges of the chains connect, in general, vertices that do not belong to the neighbor layers of $\Xi_{\tilde{m}}^n$.

Let us obtain the general description of the collection of all V-equivalence clusters of $\Xi_{\tilde{m}}^n$. If all m_i are odd, sizes of all $|\hat{\Xi}|$ subcubes of partition are equal to 2^n.

When all m_i are even, this is the case of the unique middle point, and the arbitrary vertex $\tilde{a} \in \Xi_{\tilde{m}}^n$ may have any given number $k_{\neq} \leq n$ of coordinates that are different from $m_i/2$. Volumes of subcubes corresponding to such points $\tilde{a} \in \Xi_{\tilde{m}}^n$ is $2^{k_{\neq}}$. The number of all mentioned points with $k_{\neq}$ "un-concentrated" coordinates may be calculated in the following way. It is to construct all different n-vectors that have $n - k_{\neq}$ coordinates equal to $m_i/2$ and the reminder ones accept all feasible assignments. For i and m_i number of such independent evaluations equals to $m_i/2$ taking into account that it is to consider the part of upper vertices, those from $|\hat{\Xi}|$. For one collection of fixed $n - k_{\neq}$ coordinates we receive production of terms $m_i/2$ by the set of coordinates out of the $n - k_{\neq}$ that accepted values $m_i/2$. The total number of $k_{\neq}$ different upper vertices is equal to the sum of products by all elements of $k_{\neq}$, products of terms $m_i/2$, and it involves all selections of the $k_{\neq}$ collections. Denote this number by $\hat{\varphi}(\tilde{m}, k_{\neq})$,

$$\hat{\varphi}(\tilde{m}, k_{\neq}) = \prod_{I \subseteq \tilde{m},\, |I| = k_{\neq}} \hat{\varphi}(\tilde{m}, I) = \prod_{I \subseteq \tilde{m},\, |I| = k_{\neq}} \prod_{m_i \in I} m_i/2.$$

It is easy to check that the combinatorial generating function of these numbers is

$$\hat{g}(\tilde{m}) = \prod_{i=1}^{n} (1 + (m_i/2)x). \qquad \textbf{\textit{(2)}}$$

To obtain the value of $\hat{\varphi}(\widetilde{m}, k_{\neq})$ it is to maintain the expression of $\hat{g}(\widetilde{m})$ analytically, taking the coefficient at the $x^{k_{\neq}}$.

Let us also bring the formula

$$\sum_{k_{\neq}} \hat{\varphi}(\widetilde{m}, k_{\neq}) 2^{k_{\neq}} = \prod_i (m_i + 1), \qquad (3)$$

which is some kind of check of the structural description of cube split $\hat{\varphi}(\widetilde{m}, k_{\neq})$. $\hat{\varphi}(\widetilde{m}, k_{\neq})$ is the coefficient at $x^{k_{\neq}}$ of $\hat{g}(\widetilde{m})$ in formula (2). Taking $x = 2$ in (2) we multiply the coefficients $\hat{\varphi}(\widetilde{m}, k_{\neq})$ by $2^{k_{\neq}}$, that gives the individual summand of the left side formula of (3). Substitution $x = 2$ in (2) directly, gives the right side formula of (3) proving the check.

Thus we considered two special cases of $\widetilde{m}$, one with all odd and second with all even coordinates. The general case that may have as odd as well even coordinates, may be analyzed by a simple integration of these two subcases.

Cube-Split Association Rule Mining (CSARM) Algorithm

At this point we suppose that we are given a relational table T with n attributes: $A_1, A_2, \dots, A_n$. The row collection composes a stream of transactions where Boolean and quantitative attributes are applied together. For an arbitrary categorical attribute, when it is necessary, an additional set of Boolean attributes may be generated as indicators of the different value domains of that attribute, but we suppose that the table T is already in mixed form of Boolean and quantitative attributes and it will not be changed structurally. As we mentioned above, we will consider rules of type $A \Rightarrow B$ where the left side attributes are monotonically related to the target event B. Our next supposition (secondary but important) is the notion that all transactions are sparse, which means that attributes involved with positive (none empty) values are a very limited share of the whole set of attributes. In traditional problem of supermarket basket analysis, the entire number of items in supermarket is of course very large, but each individual purchase basket contains only a very limited number of items. In these suppositions our association rule mining procedure reduces to the monotone binary function recognition algorithm.

To complete the descriptions of CSARM model, it reminds to redirect the reader to the works [Aslanyan et al., 2017] and [Aslanyan et al., 2008] that describe the Boolean monotone recognition in details. The basic application of the framework of the Cube-and-Chain Split Data Mining was with help of the new versions of Intrusion Detection system SPARTA [Aslanyan et al., 2011]. SPARTA is mining sets of LOG files in a computer system to determine the nonstandard and extraordinary behavior such as the intrusion into the system. We just need to bring the general algorithmic framework of the CSARM system.

Let $f\colon \Xi_{\widetilde{m}}^{n} \to \{0,1\}$ be a monotone function defined with the help of an oracle Ω_{f}.

CSARM algorithm at first splits $\Xi_{\widetilde{m}}^{n}$ into $\left|\hat{\Xi}\right|$ vertical equivalence classes. At second stage CSARM maps the monotone binary function f onto the constructed binary cubes. This procedure produces a large set of monotone Boolean functions that will be recognized with the help of Hansel’s algorithm.

The final step of CSARM is for integration of the fragments of recognition into a general structure of the frequent subsets of the attributes.

CSARM Algorithm.

- Partition of $\Xi_{\widetilde{m}}^{n}$ into the set of vertical equivalence classes: $V_1, V_2, \cdots, V_{|\hat{\Xi}|}$, and compose the corresponding binary subcubes: $E(V_1), E(V_2), \cdots, E(V_{|\hat{\Xi}|})$ as it is described in previous sections (cube split).
- In every cube $E(V_i)$ consider the binary function $f_i\colon E(V_i) \to \{0,1\}$, defined as follows: $f_i(\beta) = 1$ if and only if $f(b) = 1$, for every $\beta \in E(V_i)$, where b is the origin of β in $\Xi_{\widetilde{\mathrm{m}}}^{\mathrm{n}}$. f_i is monotone and is given with the help of Ω_F. Apply the Hansel’s chain split method for the recognition f_i,
- Integrate the results of $\left|\hat{\Xi}\right|$ binary recognitions procedures to obtain the target function f.

Let $\phi_A(n)$ be the minimal number of queries which is sufficient for recognizing arbitrary monotone function of n variables defined on Ξ_{m+1}^{n} by the CSARM Algorithm. Then:

$$\phi_A(n) = \sum_{k=0}^{n}\left(\hat{\varphi}(\widetilde{m},k)\cdot\left(C_k^{\left\lfloor\frac{k}{2}\right\rfloor} + C_k^{\lfloor k/2\rfloor+1}\right)\right) \text{ for all } m_i \text{ even, and}$$

$$\phi_A(n) = \prod_i\left(\frac{m_i+1}{2}\right)\cdot\left(C_n^{\left\lfloor\frac{n}{2}\right\rfloor} + C_n^{\lfloor n/2\rfloor+1}\right) \text{ for all } m_i \text{ odd.}$$

Thus we obtained formulas for $\phi_A(n)$ in two special cases of $\widetilde{m}$, one with all odd and second with all even coordinates. The general case that may have as odd as well even coordinates, may be analyzed by a simple integration of these two subcases.

Bibliography

[Agrawal et al., 1993] Agrawal R., Imielinski T., and Swami A.. Mining association rules between sets of items in large databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington D.C., May 1993.

[Agrawal et al., 1994] Agrawal R., Srikant R.. Fast algorithms for mining association rules. Proceedings of 20th International Conference Very Large Data Bases, VLDB, Santiago, Chile, pp. 487-499.

[Agrawal et al., 1996] Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo I.. Fast Discovery of Association Rules. Advances in knowledge discovery and data mining 12 (1), pp. 307-328, 1996.

[Armstrong, 1998] Armstrong T., Marriott K., Schachte P., Sondergaard H.. Two Classes of Boolean Functions for Dependency Analysis. Science of Computer Programming, 31(1): 3-45, 1998.

[Aslanyan et al., 2017] Aslanyan L., Sahakyan H.. The Splitting technique in monotone recognition. Discrete Applied Mathematics, 216 (2017), pp. 502–512.

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

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

[Aumann et al., 2003] Aumann Y., Lindell Y.. A Statistical Theory for Quantitative Association Rules. Journal of Intelligent Information Systems, vol. 20, 255-283, 2003.

[Boulicaut, 2001] Boulicaut JF., Bykowski A., Jeudy B.. Towards the Tractable Discovery of Association Rules with Negations. In Larsen H.L., Andreasen T., Christiansen H., Kacprzyk J., Zadrożny S. (eds) Flexible Query Answering Systems, Advances in Soft Computing, vol 7, 2001, Physica, Heidelberg.

[Fukuda et al., 1996a] Fukuda T., Morimoto Y., Morishita S., Tokuyama T. Mining Optimized Association Rules for Numeric Attributes. PODS '96 Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pp. 182-191, Montreal, Quebec, Canada, June 04 - 06, 1996; Journal of Computer and System Sciences, Volume 58, Issue 1, pp. 1-12, February 1999.

[Fukuda et al., 1996b] Fukuda T., Morimoto Y., Morishita S., Tokuyama T.. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. Proceedings of the ACM SIGMOD Conference on Management of Data,' pp. 13-23, June 1996.

[Hahsler, 2017] Annotated Bibliography on Association Rule Mining by Michael Hahsler, http://michael.hahsler.net/research/bib/association_rules

[Han et al., 2007] Han J., Cheng H., Xin D., Yan X., Frequent pattern mining: Current status and future directions, Data Mining and Knowledge Discovery, 14(1), 2007.

[Hansel, 1966] Hansel G.. Sur le nombre des functions booleennes monotones de n variables, C.R. Acad. Sci. Paris, 262, serie A (1966), 1088.

[Iberman et al. 2001] Imberman S., Domanski B., Finding Association Rules from Quantitative Data Using Data Booleanization, Americas Conference on Information Systems, Proceedings, 2001.

[KOR,1965] V. Korobkov, On monotone functions of algebra of logic, Prob. Cyb. 13 (1965).

[Li et al.] Li H. F., Lee S. Y., Shan M. K.. DSM-PLW: single-pass mining of path traversal patterns over streaming web click-sequences. Proc. of Computer Networks on Web Dynamics, pp. 1474–1487, 2006.

[Mannila et al., 1994], Mannila H., Toivonen H., Verkamo I., Efficient algorithms for discovering association rules, In AAAI Workshop on Knowledge Discovery in Databases, pp. 181-192, Seattle, Washington D.C., AAAI Press, 1994.

[Piatetsky-Shapiro, 1991] Piatetsky-Shapiro G., Discovery, analysis, and presentation of strong rules, In Knowledge Discovery in Databases, pp. 229-248, 1991.

[Rastogi, 2002] Rastogi R., Shim K., Mining Optimized Association Rules with Categorical and Numeric Attributes, IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 1, 2002.

[Salleb-Aouissi et al., 2007] Salleb-Aouissi A., Vrain C., Nortet C.. QuantMiner: A Genetic Algorithm for Mining Quantitative Association Rules. Proceedings of the 20th International Conference on Artificial Intelligence, 2007, pp. 1035-1040, India, http://www1.ccls.columbia.edu/~ansaf/QuantMiner

[Srikant et al., 1996] Srikant R., Agrawal R.. Mining Quantitative Association Rules in Large Relational Databases. Proc. of ACM SIGMOD Conference on Management of Data, Montreal, Canada, 1996.

[Toivonen, 1996] Toivonen H., Sampling Large Databases for Association Rules, In Proceedings of the 22th International Conference on Very Large Data Bases, pp. 134-145, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

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

[Yoda et al., 1997] Yoda K., Fukuda T., Morimoto Y., Morishita S., Tokuyama T., Computing optimized rectilinear regions for association rules, in Proceedings of the Third International Conference on Knowledge Discovery and Data Mining,' pp. 96-103, Aug. 1997.

[Webb, 2001] Webb G. I. Discovering associations with numeric variables. Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 383-388, New York, 2001, ACM Press.

[Aslanyan et al., 2011] L. Aslanyan, H. Sahakyan et al, Managing Risk and Safety (chapter 1), Intelligent Data Processing in Global Monitoring for Environment and Security, ITHEA, Sofia (Bulgaria) and Kiev (Ukraine), Editors: K. Markov and V. Velichko, ISBN: 978-954-16-0045-0 (printed), 410 p., 2011.

Authors' Information

Levon Aslanyan *– Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia, head of department; 1 P.Sevak str., Yerevan 0014, Armenia; e-mail: lasl@sci.am*

Major Fields of Scientific Research: Discrete analysis – algorithms and optimization, pattern recognition theory, information technologies.

Hasmik Sahakyan *– Institute for Informatics and Automation Problems of the National of Science of Armenia; Scientific Secretary. 1 P.Sevak str., Yerevan 0014, Armenia; e-mail: hsahakyan@sci.am*

Major Fields of Scientific Research: Combinatorics, Discrete tomography, Data Mining