A Randomization Test for extracting Robust Association Rules

Martine Cadot

Title: A Randomization Test for extracting Robust Association Rules

Author : Martine Cadot, Université Henri Poincaré/LORIA, Nancy (France), Martine.Cadot@loria.fr

Abstract :

An association rule "if A then B" is a link between database property sets A and B. Since this type of rule is not deduced from hypotheses, but found by investigation in data, association rules extraction belongs to Data Mining techniques (Han et al. 2001). Presently, more than fifty different measures are used to try to establish the quality of association rules, according to their different semantics. It shows the great variety of links between properties expressed by these rules, but also the difficulty of being sure they are meaningful. To test if an association rule is robust, that is to say to determine if the link it brings out is not due to chance, a Randomization Test (Edgington, 1995) is developed. For this, simulations that allow the generation of numerous artificial databases identical to an original database, except for the links between properties, are defined. Only the links which are found in the original database and in less than 5% of the artificial databases are judged statistically significant, with a type I error risk of less than 5% (Snedecor et al., 1967), and produce significant association rules. This simulation technique is far more efficient than the acceptance-rejection method and allows the use of the associated randomization test in various databases.

Keywords: statistics, data mining, quality measure, association rule, threshold, itemset, simulation, randomization test, permutation test.

1. Introduction

Data mining techniques like association rules extraction can be used to analyse large amounts of data (Hand et al. 2001). The most widely known association rule was extracted from US market basket data composed of transactions (lists of items bought in a single purchase by a customer). It is "x buys diapers → x buys beers" (Han et al. 2001). Clearly, not all the customers who buy diapers also buy beers. But the proportion of customers that verify this surprising rule is sufficiently high to interest supermarket managers. Presently, association rules extraction is not used to mine only marketing databases. It is an efficient data mining technique for extracting knowledge from most databases, for example corpuses of scientific texts as in the proposed application. To make sure they produce worthwhile knowledge, researchers have already defined more than fifty quality measures for association rules (Guillet, 2004). These measures are constructed to sort rules with different semantic approaches (Briand et al., 2004). Usually, arbitrarily fixed thresholds of some measures are used to separate meaningful association rules from others. But, besides the difficulty of choosing and combining different measures and their thresholds, chosen thresholds are generally increased afterwards and even supplementary measures added to decrease the number of extracted rules, which is too high for interpretation by an expert. To avoid the pruning of meaningful rules which results from this hazardous process, navigation tools on numerous rules with common measures have been elaborated (Botta et al. 2002). In this paper, a different method with the same aim of limiting the effects of this risky process of pruning is presented. The method allows to separate accidental rules from generalizable rules. It belongs to the family of statistical tests, which separate out significant data links from accidental ones, but not to the “Test theory” of classical statistics (Snedecor et al., 1967). In fact, classical statistics work badly with numerous Boolean properties (Cadot et al.

2005). Moreover, to use classical tests it is often necessary to know the data probability laws, or to accept approximate data normality, whereas a test that works with data from all origins with various probability laws is called for. A non-parametric test method (Hollander et al., 1973) is a former alternative to the classical test method, which allows to neglect the data probability laws, but has other restrictions. The method chosen in this paper belongs to randomization tests (Edgington, 1995), a new alternative statistical methodology which works with all data types. This methodology does not neglect data probability laws, but by use of random artificial data, withdraws them from data. In the same process, it withdraws all irrelevant data features, and keeps only the feature under consideration. This process of artificial database generation is independently repeated a hundred or more times, and the presence of the feature in consideration is sought in each artificial database. If it is found with a relative frequency lower than an alpha threshold (of 0.05 for example), the conclusion is that one can be almost sure (with an error probability of 5%) that the feature is not due to chance, but significant. Otherwise the feature is attributed to chance. With this methodology, the simulation technique associated to a randomization test is crucial to the construction of artificial databases able to keep and withdraw specific features of the original database. A randomization test is constructed that will be used to determine if the links between numerous Boolean attributes expressed by “frequent itemsets” (sets of properties which are often enough simultaneously verified) are accidental or not, with a risk error lower than a fixed value (usually, alpha=.05).

To start this paper, a short presentation of association rules (ARs) extraction principles highlights the deficiency of robustness which requires correction by an appropriate statistical test. Then in the next parts, the construction of a randomization test is presented that separates significant rules from others. First, the framework of this test, ARs extraction and statistical methods, are presented to justify the needs of a randomization test for robust ARs extraction. Second, the

requirements of the simulation method of this test are defined. It must be able to draw off from a database only "context-free" links, which are the most adapted to robust ARs, among all the links simply opposed to independence. The need to divide database links into co-occurrence type and context type is justified by means of a real database. Third, the simulation technique features are detailed to generate easily and rapidly numerous artificial databases, which respect the above requirements. Fourth, the simulation technique of the randomization test is compared with the acceptance-rejection method. The last part concludes and gives some perspectives.

2. Definition of an AR

The definition of an AR (such as A→B) has its origin in three principal disciplines. O1) In didactics, with Gras (1979), A and B are learning items (such as the capability of solving an equation system, or translating a problem into an equation system) and for each pupil or listener, results of evaluation allow to establish if each item is known or not. O2) In the humanities, with Guigues et al. (1986), A and B are symptoms of disease (such as depression, or anxiety) and for each patient, the psychologist indicates if a symptom is present or not. O3) In marketing, Agrawal et al. (1996), A and B are market items (such as diapers and beers) and one can find in each checkout ticket if these items have been bought or not. So, according to the three authors, A and B belong to a set of binary properties (acquisition of a capability, presence of a symptom, purchase of a market item) whose value (True/False) is known for each object of a given database (pupil, patient or transaction) and the AR A→B is extracted from data only if it is a meaningful rule. To be sure that A→B is a meaningful rule, one can require that the number of objects simultaneously verifying properties A and B is high (O1 and O3), this number is named "support". It is also possible to require that the proportion of objects, having property A without

property B, be low (O3 using the "confidence" index), be statistically surprising (O1 using the "intensity implication" index), or be naught (O2). Moreover, all the meaningful rules between properties have to be found. An AR with more than two properties, such as ABC→DE, is defined by O3 as a rule A'→B' where A' is the conjunction of properties A,B,C (named an "itemset") and B' of D, E, and can be extracted as easily as rules with two properties, because of adapted algorithms. But for O1 and O2, fusion is not achieved between properties but between rules, by inference rules such as transitivity for example.

Presently, the three approaches are unified in a methodology of ARs extraction. With fast algorithms (Bastide 2002), all the itemsets of support greater than an arbitrarily chosen threshold are found, and then more than fifty indices of quality (Guillet 2004) help choose meaningful ARs among all those constructed by partitioning itemsets into two parts (If an itemset contains n properties, it gives 2^n-2 potential ARs). A little example of how to find all the ARs by the most common method illustrates this.

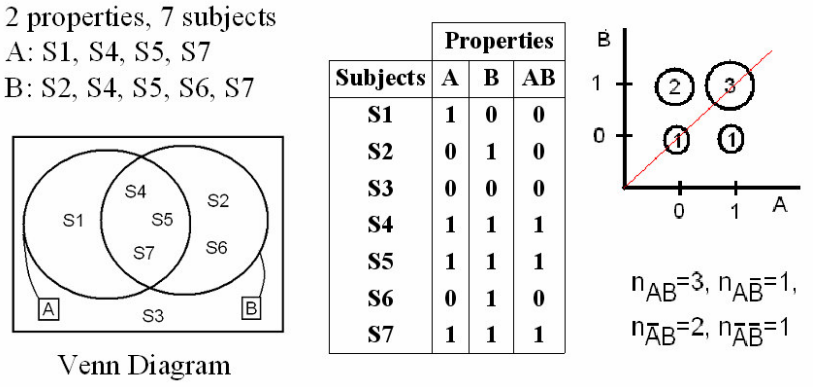


Figure 1: Extraction of A→B from data where objects are individuals named subjects.

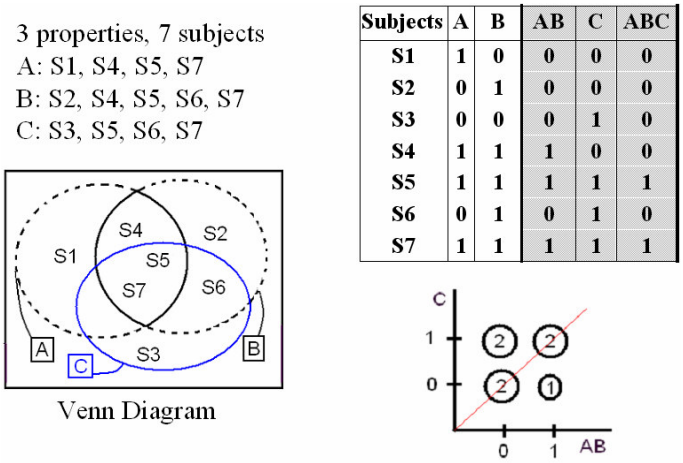
a. Three principal forms of data can be given for ARs extraction: a list of properties verified by each object, such as a checkout ticket, a list of objects which verify each property, such as in the given example (fig. 1, top left), or a Boolean table (True: 1, False: 0) with

properties in columns and objects in rows as part of a table with properties A, B (fig. 1, middle).

b. The itemset AB is constructed as the intersection of sets A and B in a Venn diagram (fig. 1, bottom left), or by the fusion of columns A and B in column AB, the fusion operator being the product of Boolean values in a table (fig. 1, middle). Three subjects, s4, s5 and s7, verify AB, so the support of AB is 3. If this support is found high enough (for example if the threshold of the support is 1), this itemset is kept for later AR extraction.

c. If the itemset AB is kept, two rules are candidates, A→B and B→A. For each rule, quality measures are computed to determine if the rule is kept or not. To compute most of these indices, only the four numbers (fig. 1, bottom right) are necessary, among them three need to be calculated (the first of them is known). For example, the confidence index is nAB/nA, 0.75 for A→B and 0.6 for B→A. Depending on the confidence threshold (for example 0.5, 0.7, 0.8), both rules are extracted, or only rule A→B, or no rule at all. These two indices, support and confidence, are quality measures coherent with a representation of the rule A→B as B=f(A) (fig. 1, top right): B increases with A and the counterexample number of rule A→B, appearing below the diagonal, is 1, which is low. If this diagram is turned upside down, it represents rule B→A as A=f(B) whose quality is smaller having two counterexamples.

For more than two properties, candidates for itemsets and ARs are numerous, so fast algorithms (Bastide, 2002) are used to find them all. But found itemsets and ARs are pruned with the same method. This can be seen in figure 2 for AR A→BC extraction. The data are the same as in figure 1 except for the addition of property C.



Subjects	A	B	AB	C	ABC
S1	1	0	0	0	0
S2	0	1	0	0	0
S3	0	0	0	1	0
S4	1	1	1	0	0
S5	1	1	1	1	1
S6	0	1	0	1	0
S7	1	1	1	1	1

Figure 2: Extraction of AB→C from data.

An example has served to show the most common method of ARs extraction, but confidence in c) can be replaced or completed by one or several of the other quality measures, according to the semantic choice of the expert in interpreting ARs (Guillet, 2004). The difficulty is choosing among all these measures, and fixing their thresholds, with the consequence of producing different ARs sets. Only choosing the support and confidence indices is rarely sufficient. To illustrate the necessity of a sensible pruning choice, if ARs are computed from House-vote Data (downloaded from UCI Repository http://www.ics.uci.edu/~mlearn/MLRepository.html) under the form of 50 binary properties (dichotomisation of the 23 original qualitative properties) and 435 subjects, with a threshold of 40 for support and 0.9 for confidence, more than 1.8 million ARs with a number of properties between 2 and 14 are obtained. In this paper, the aim is not to contribute to AR pruning at stage c), but to delete all accidental itemsets found at stage b) before executing stage c). The next part discusses how non-accidental itemsets can be separated from accidental ones.

3. Choosing a test type for bringing out strong links between properties

In stage b), itemsets are kept with high supports to bring out the strongest links between properties. These links are of the co-occurrence type, opposed to independence. So by choosing a high enough threshold for support, one can expect that properties in an itemset whose support exceeds this threshold are not independent. But two itemsets, such as AB and CD, can have equal supports with independence between A and B, and a strong link between C and D, if A and B are common properties in the database and C and D rare ones. Then the choice of an arbitrary support threshold, the same for each itemset, can be justified to run the algorithms of stage b) faster, but not to extract the only meaningful itemsets. To obtain for each itemset a support threshold that guarantees the itemset against independence of its properties, a statistical test for itemset support is constructed.

A randomization test (Manly,1997, Edgington, 1995) is chosen, which allows the testing of data links when probability laws are difficult to establish. For example, to assume a correlation between two variables with a probability $p<0.05$ to fail, one has to establish normality of variables because the calculation of the threshold depends on probability laws (Snedecor et al., 1967). If one permutates randomly the values of the first variable, they become independent, and if this permutation is repeated independently many times, one obtains the correlation law when the variables are independent, and the threshold for $p<0.05$ is easy to deduce (Manly, 1997). Randomization tests belong to distribution-free tests. The Fisher test, which replaces the Chi2 test (Agresti 1992) for contingency tables where some frequencies are too small, can been seen as an ancestor of randomization tests in the classical statistical testing theory (Edgington, 1995). The randomization tests require intensive computer use, so they have waited for computation

capacities to increase. They are largely used in ecology (Legendre et al., 1998). They are particularly well adapted to this problem of finding a support test, which is valid for databases of various origins, and in a more general way, to robust data mining. An important stage of this test construction is the definition of the null hypothesis by adapted simulations.

A randomization test being chosen for robust ARs extraction, the construction of an associated simulation method depends on the choice of a null hypothesis, which is often written in the same form as that imposed by classical test theory, then translated into an algorithm for computation. But the definition of the null hypothesis for constructing a test adapted to itemset supports in a database is problematic under classical statistical theory. The common, first stage is to establish the probability laws of all the AB itemset supports under the null hypothesis of independence of A and B whatever database properties A and B may be. So, the null hypothesis is not a simple hypothesis for two properties, but a repeated hypothesis for all the pairs (or greater sets) of database properties. It has been noted that scientific researchers propose methods to transform tests working with a simple hypothesis into tests working with their composite versions. Their adaptations depend on the specificity of the test and on the different points of view, (for example in the linear model, the alpha risk can be reduced in several ways in Howel [2000]; the properties can be fixed or randomly chosen in Winner [1991]; the contrasts can be a priori or a posteriori in Howel [2000]). This form of null hypothesis definition seems to bring about a loss of robustness for this test and it seems preferable to choose an algorithmic formulation of the null hypothesis closer to the simulation methodology of randomization tests. The next part specifies the choice of these two elements for the itemset support test.

4. Choice of the simulation method for bringing out strong links between properties

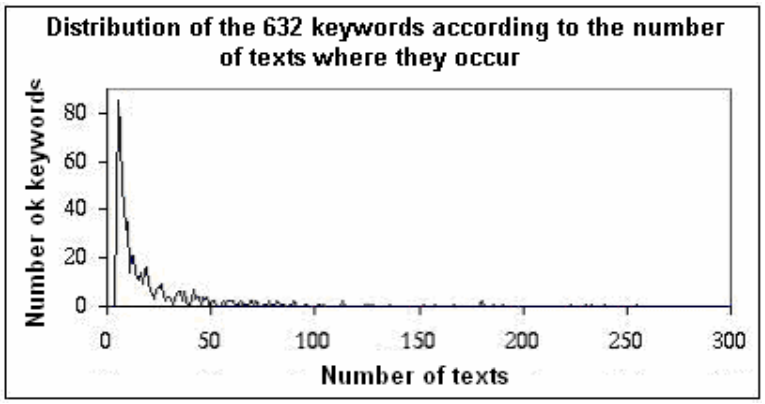


Figure 3 : Distribution of keywords in the corpus of texts.

A similar method to the one described in a preceding paragraph for the correlation test is first tried out on a database with 632 keywords that belong to keyword sets from 1361 texts of a scientific domain (INIST, Nancy). This database was chosen for two reasons. The first is that textual data have particular probability laws. In figure 3, one can see that these data are not distributed following the Laplace-Gauss law. Particularly, the Estoup-Zipf law (Guiraud 1960) is known in statistical linguistics for printed matter and the "Zipf-like" law for web mining (Breslau 1999).

These text data are of the latter type, as can be seen in fig. 4, so if the tests are not really “distribution-free”, but only accept a little difference with the commonly used laws in these cases, such as the Poisson law, the binomial law or the Laplace-Gauss law, surely they are apt to fail on these data. The second reason is that strong links have been found between the keywords that produces ARs of high value for quality measures such as support and confidence, considering the texts as objects and the keywords as properties. So after deleting all the accidental itemsets, still numerous itemsets must remain.

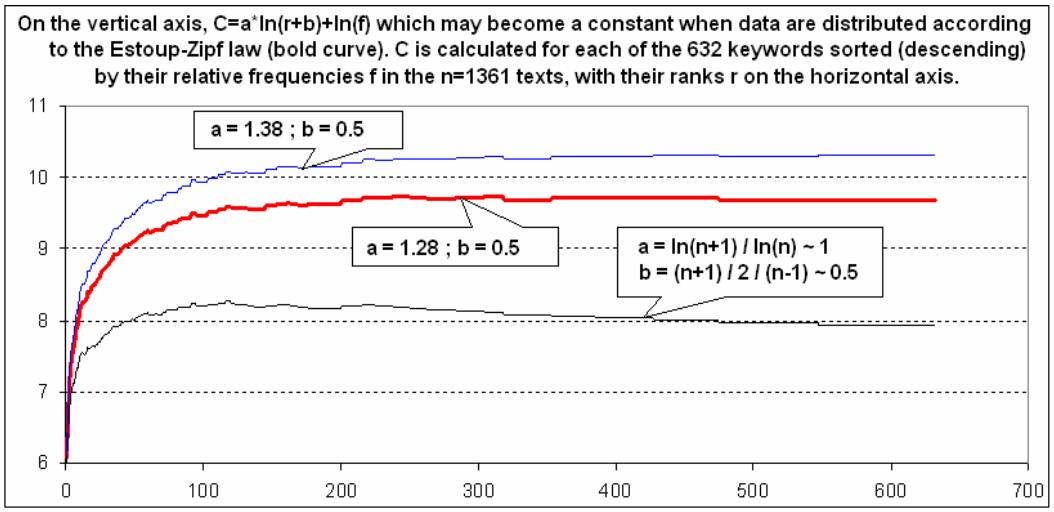


Figure 4 : Bringing out the parameters (a=1.28 and b=0.5) of the "Zipf-like" law of the corpus of texts.

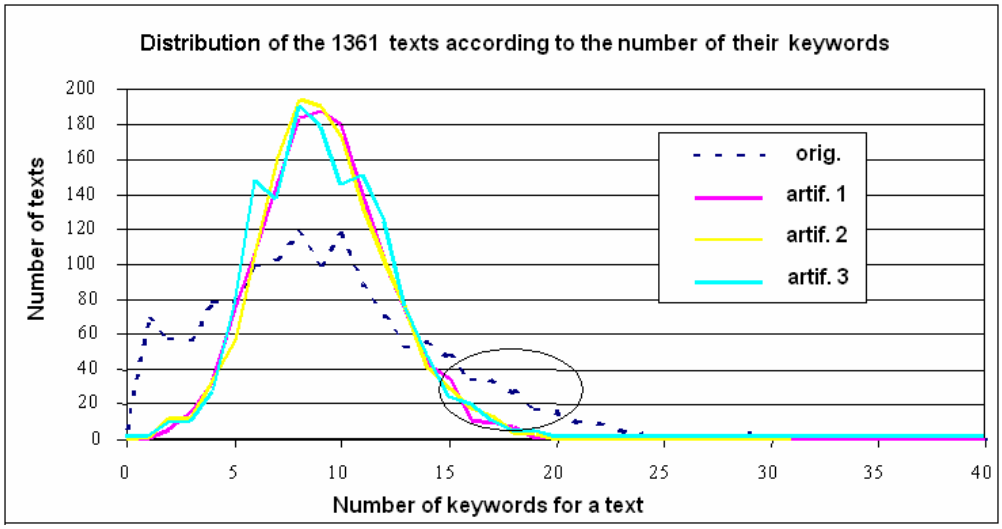


Figure 5: Differences between the original data and the artificial data for the first trial run of simulations

The first trial run is a set of independent simulations of data. Each keyword is put in the same number of texts as it was found in, in the original data, but these texts are randomly chosen. In

figure 5, three artificial databases and the original database are represented in the same graph. But a comparison of the original database with the artificial ones shows that the global characteristics of this database are modified by these simulations: as indicated by the black ellipsis for example, by these simulations less texts with about twenty keywords are obtained than in the original database, and generally, less texts with many words and also less texts with few words, and more texts with a number of keywords between 6 and 13. This simulation type modifies the data structure. If this simulation method is used to generate artificial databases, the significativity of an itemset support, obtained by confrontation of the original database to artificial ones, can be imputed to two effects, the original database structure and the co-occurrence link between its properties in the original database. There is no possibility of being sure that it is only the latter, which is worthwhile for robust ARs extraction.

Other simulations have been tried out to respect the database structure. Only the simulations which conserve the marginal sums of data, that is, keep constant the number of keywords per text and the number of occurrences in the texts per keyword, have succeeded in respecting this database's structure. Later how these simulations can be carried out by adapted permutations will be described, but the last trial run results, which meet expectations, are specified now. With an α risk of less than 5% of finding a link between two properties A and B of the original database where they are independent (this risk is named the type I error risk [Snedecor et al., 1967]), the threshold of the support of their itemset (AB) is chosen equal to a value which is exceeded by only five of all the supports of this itemset in the artificial databases obtained by one hundred independent simulations. If the support of AB in the original data is higher than this threshold, itemset AB is deleted. The itemsets with more than two properties are deleted in two different

cases: in the first case because they exceed their threshold calculated with the same method as above, and in the second case because they contain properties composing an itemset deleted at a preceding stage. And each AR whose properties compose a deleted itemset is deleted. As can be seen in table 1, with the last method more accidental itemsets are found than with the first. This supplementary deleting can be interpreted by the suppression of the links due to the specific data structure, such as the Zipf-like law. Moreover, the rare itemsets deleted in the first trial run, but not in the last one, exceed their last trial run thresholds by a few units, which can be attributed to sample fluctuations inherent to random based methods. This would seem to imply that these simulations form a coherent method.

Number of properties in itemsets	2	3	4	5	Total
Number of itemsets with support>=10					
before deleting	1157	424	86	14	1681
deleted with first trial run	338	110	16	1	465
deleted with last trial run	475	216	41	6	738
Number of rules with confidence>=0.8					
before deleting	58	217	172	64	511
deleted with first trial run	0	33	31	6	70
deleted with last trial run	0	58	70	28	156

Table 1: Some results of the pruning of the itemsets and the ARs obtained with the two simulation types

With this last simulation method, the links between the original database properties are deleted in the artificial databases without changing the marginal sums, so the links which are the consequence of the marginal laws of the original database are kept, and the artificial databases have the same structure as the original database, the only difference is that the properties of these

artificial databases are independent (and the objects too, by duality). Consequently, the last simulation type allows the testing for database properties independence given the database's global structure. The need to separate co-occurrence links from data structure links are not unique to textual data, but can be extended to the databases obtained by observations or surveys. For example, in psychological questionnaires, some patients have a tendency to declare too many symptoms and others too few, and some symptoms are rare and some frequent, so there can appear to be co-occurrence of two rare symptoms which is only the effect of the first patient type. With the simulations of the preceding paragraph, which conserve only the number of patients having each symptom, this link cannot be separated from the dependence link, but with simulations which conserve also the number of symptoms declared by patients, it can be separated. So, this simulation method allows an efficient pruning of itemsets by deleting those which represent the links due to chance and those which represent the links due to data structure.

This simulation can be made with several methods of the acceptance-rejection type. For example with the one consisting in randomly permutating the values of each column, and keeping only the resulting databases with the row sums identical to the ones of the original data. But with this method, it is necessary to compute a huge number of artificial databases to obtain one which conserves the marginal sums of the original data. So a simulation method is proposed in which each generation run produces an artificial database with the correct marginal sums. This method is described in the next part.

5. Generation of all the Boolean arrays with the same marginal sums of a given Boolean array.

Let M be a relation between two Boolean arrays of same dimension n x p (n the number of rows, p the number of columns) defined by: B1 and B2 are in relation M if their marginal sums are

identical. Clearly, this relation is an equivalence relation in the set of all Boolean arrays of dimensions equal to n x p. Let B1 be the Boolean array corresponding to the original data, with p properties and n objects. It must be possible to obtain each of the M-equivalent to B1 arrays by the simulation process of the randomization test, with a given probability. To be sure that a simulation process can find all the Boolean arrays, it is necessary to be sure that for any two Boolean arrays B1 and B2 of dimensions n x p, a simple transformation (or a composition of simple transformations), which maps B1 on B2, can be found. The transformations proposed are of two types, named “rectangular-exchange” and “cascade-exchange”. In this section and in three stages, why they are convenient to the simulation process is established. First, their definitions are given, second, their property of generating all the M-equivalent arrays of a Boolean array by composition is shown, and third, their ability to be replaced by compositions of only rectangular-exchanges is shown. Probabilities are discussed in the next part.

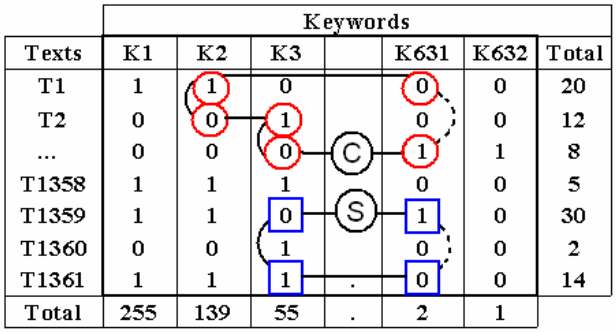


Figure 6: C (red circles) is an cascade-exchange. S (blue squares) is a rectangular-exchange

In figure 6 a part of the Boolean array B1, which corresponds approximately to the text database, is represented. In the columns are the keywords, sorted by their supports (marginal sums) and in the rows are the texts. If value 1 is in the cell at the intersection of a row and a column, it means the corresponding keyword is present in the list of the corresponding text keywords and absent if the cell value is 0. For example, the keyword K2 is in the keyword list of text T1, but not of text

T2. Let C be the transformation, which replaces all the binary values inside red circles by their 1-complement (0 becomes 1-0=1, and 1 becomes 1-1=0), and let S be the one that transforms, in a similar manner, all the binary values inside blue squares. C is named a “cascade-exchange”, because the mapping of C on B1 produces serial exchanges by putting the 0 of cell (K361, T1) into cell (K2, T1) and takes the 1 of the latter cell to cell (K2, T2), and so on, like a cascade indicated by black lines (it can be necessary to re-order rows to obtain this cascade which runs from the top to the bottom), except for the last cell, whose value is put into the first one (dotted line). These six successive exchanges alternately work in a column or in a row and keep unchanged the marginal sums of the rows and columns concerned. The length of a cascade-exchange is the number of rows or columns concerned, and also the number of zeros and ones which are exchanged, so the length of C is 3. S is named a “rectangular-exchange” because the cells that it changes are in the vertices of a rectangle. It can be considered as a cascade-exchange of length 2. With the mapping of C (see figure 7), the Boolean array B1 becomes array B2 and with the mapping of S on B2, it becomes array B3, these three arrays being M-equivalent, because S and C keep marginal sums. Inversely, the exchanges of values allowing the transformation of B1 to B3 can be written as a composition of two exchanges, one C type and the other S type, but it is less easy to discover. Proof is too long to appear in this paper, but details of proof principles are shown in the example in the next paragraph.

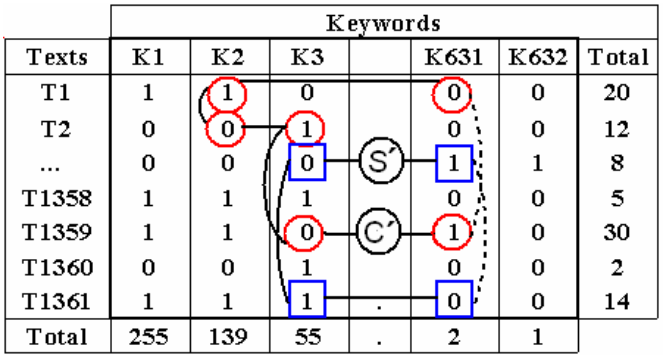
	K1	K2	K3		K631	K632
T1	1	1	0	.	0	0
T2	0	0	1	.	0	0
...	0	0	0	.	1	1
T1358	1	1	1	.	0	0
T1359	1	1	0	.	1	0
T1360	0	0	1	.	0	0
T1361	1	1	1	.	0	0
Total	255	139	55	.	2	1

K1	K2	K3		K631	K632
1	0	0	.	1	0
0	1	0	.	0	0
0	0	1	.	0	1
1	1	1	.	0	0
1	1	0	.	1	0
0	0	1	.	0	0
1	1	1	.	0	0
255	139	55	.	2	1

K1	K2	K3		K631	K632	Total
1	0	0	.	1	0	20
0	1	0	.	0	0	12
0	0	1	.	0	1	8
1	1	1	.	0	0	5
1	1	1	.	0	0	30
0	0	1	.	0	0	2
1	1	0	.	1	0	14
255	139	55	.	2	1	

Figure 7: on the left B1 which, with C, becomes B2 in the middle, and, with S, B3 on the right.

Now, B1 and B3 are given, and their non-identical values are browsed to find exchanges whose composition transforms B1 to B3. Let us suppose we start by the value 0 of B1 (see fig. 7) in cell (K3, T1359) in a blue square, which has changed to 1 in B3. Because the sum of column K3 is unchanged (its value is 55 for B1 and B3), we know there is an inverse change (1 to 0) between B1 and B3 in a cell of this column. We find it in cell (K3, T1361) or (K3, T2). Then, let us suppose that we choose (K3, T1361), in a blue square. Now we are in row T1361 with the same marginal sum (14) for B1 and B3, so we know that we can find a cell with the inverse change (0 to 1) on this row. It is in cell (K631, T1361). When we look for a cell with an inverse change (1 to 0) in column K361, we choose (K361, T1359), because T1359 is the row of the departure cell, and we have found S. There remains only cells in red circles, whose values differ between B1 and B3, so if one of these is chosen, with the same method as above, S is found. If we had made other choices, we would have also obtained S and C or S' and C', where S' is a rectangular-exchange and C' a cascade-exchange of length 3, as can be seen in figure 8. So, the decomposition of a transformation which respects M-equivalence is not exactly unique, but all the decompositions have a minimal number of exchanges of each type, here one rectangular-exchange and one cascade-exchange of length 3. Given two M-equivalent arrays, a minimal exchange set, whose composition transforms one of the two into the other, can be found by using the algorithm that has been described in this paragraph. The next paragraph shows how only rectangular-exchanges can be used for this transformation.



	Keywords						
Texts	K1	K2	K3		K631	K632	Total
T1	1	1	0		0	0	20
T2	0	0	1		0	0	12
...	0	0	0	S′	1	1	8
T1358	1	1	1		0	0	5
T1359	1	1	0	C′	1	0	30
T1360	0	0	1		0	0	2
T1361	1	1	1	.	0	0	14
Total	255	139	55	.	2	1	

Figure 8: The other decomposition of the transformation that maps B1 on B3.

A cascade-exchange of length L can be decomposed into L-1 rectangular-exchanges (proof is too long for this paper but its principles are shown in the example of fig. 9). C has length 3, and 6 cells, so can be decomposed into two rectangles with exactly one common vertex. When the two rectangular-exchanges corresponding to these rectangles are successively mapped, the value of the common vertex changes twice, so it is the same in B1 as in B2. The cell in this common vertex is named “pivot”. In fig. 9, a cell (K3, T1) is chosen as the pivot. So, the first rectangle corresponds to the rectangular-exchange T2, and after mapping T2, the pivot value is 1, and the rectangular-exchange T3 can be mapped. So the transformation C is the composition of T2 and T3. This decomposition into two rectangular-exchanges is not unique. It can be done with the pivot in cell (K2, T3) or (K631, T2). Because of the common cell, this decomposition is not commutative, contrary to the decomposition of the transformation into S and C, which have no common cell. To conclude, it is possible to transform B1 into B3 with a composition of three rectangular-exchanges, for example T1, T2 and T3 as in the fig. 9. So, if on B1 all possible successions of all the possible rectangular-exchanges are mapped, all the M-equivalent to B1 arrays are obtained. It can be concluded that all the M-equivalent arrays of a Boolean array can be obtained by composition of the rectangular-exchanges. Given B1, to generate a random M-equivalent array, it is sufficient to choose a random number (k) of exchanges, and successively,

and independently choosing k random rectangular-exchanges. But these choices can be made in several manners. A manner that respects the distribution of the artificial databases may be chosen. It is the aim of the next part.

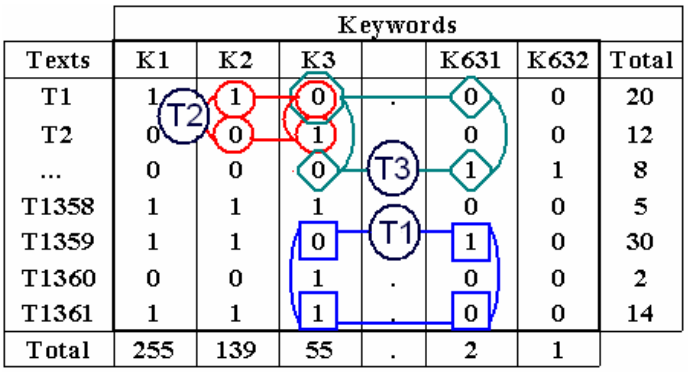


Figure 9: Transformation of B1 to B3 is the composition of T1, T2 and T3

6. Comparison of the two simulation methods: rectangular-exchange and acceptance-rejection

Rectangular-exchanges can be used in a simulation method in different manners, and for different parameter values. They must be optimized in either of two different ways: the observed frequencies of the artificial databases must be close to their expected frequencies if they are greater than 1, and if they are less than 1, the variety of artificial databases must be large. The rectangular-exchange simulation method is successively adapted to these two cases, and the comparison is made between it and an acceptance-rejection method.

To create an artificial array, there are several successive and similar stages. In a stage, four numbers, two for rows and two for columns are randomly and independently chosen. If the corresponding rectangle can be associated to a rectangular-exchange, the values are exchanged, and nothing is done if it cannot. In figure 10, the horizontal axis represents the number of

successive stages used in a trial run of 1000 simulations, and on the vertical axis the corresponding frequencies of each artificial array which is M-equivalent to the original array. The original array is on the top left. And above each of the five M-equivalent arrays, there is a different symbol (such as a blue lozenge for the artificial array identical to the original array), which allows the differentiation of the curve of frequencies of each artificial array in the simulated data.

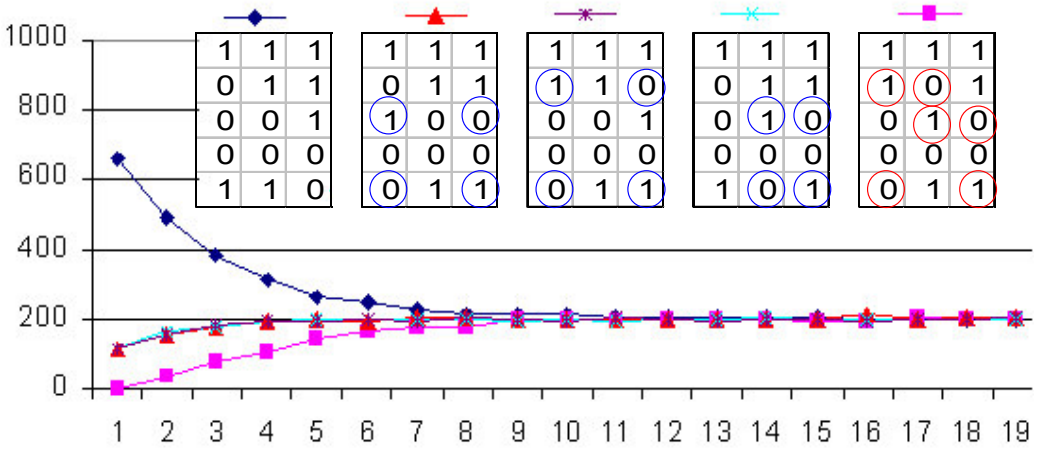


Figure 10: Result of 1000 simulations: the coordinates are the number of apparition of each array and the number of successive exchanges in a permutation.

It can be seen in figure 10 that for less than 10 stages (left of curves), the frequencies of the array on the top left is greater than expected (the curve with blue lozenges), and inversely for the array on the top right (the curve with pink squares). For the first array, identical to the original array, it is because with the composition of even numbers (which can be zero) of any identical rectangular-exchange, the original data are again obtained, and for the last, because it can be obtained by the mapping of the cascade-exchange of length 3 on the original data (red circles in figure 10), which can be obtained only if particular rectangular-exchanges are made in a particular order in these stages. To obtain each of the three other arrays, one particular

rectangular-exchange is sufficient. So, to obtain approximately the same number of each of the five M-equivalent arrays, as expected in accordance with the probability calculus, at least ten stages have to be run. Other choice types converging with less than 10 stages were tried, for example exactly one rectangular-exchange per stage (instead of 0 or 1), but most of them produced biases, the differences between the observed frequencies and the expected frequencies of some arrays remaining constant after convergence. However, no bias has been found when running each of 1000 simulations not on the original array but on the array obtained by the preceding simulation. So, if the number of stages is fixed at 10, with the above explained method, there are 10 stages for each simulation, and with the new method, 10 stages for the first simulation, 20 for the second and so on. This change decreases the number of stages necessary to converge. So this second version of the rectangular-exchange method is now used. Nevertheless, the number of stages cannot be reduced to 1, as can be seen in figure 11.

1	0	0	0	0
1	0	1	0	0
1	0	0	1	1
1	1	1	1	0
1	1	1	0	1
0	0	0	1	1
0	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

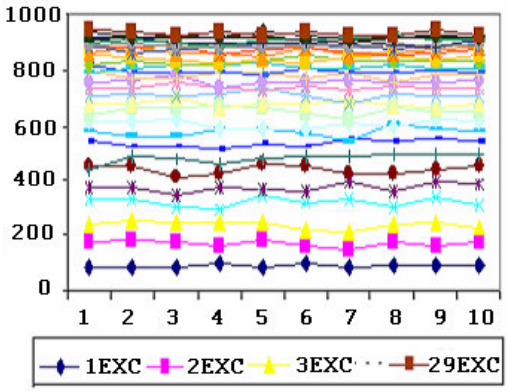


Figure 11: Results obtained for data on the left for 10 trial runs of 1000 simulations with a number of exchanges between 1 to 29.

Now a check is made to ensure that it is possible to choose the parameters (here the number of stages) of the rectangular-exchange simulation method so as to have a great number of different

artificial databases in case their expected frequencies are less than 1. An example of an original database that verifies this property is on the left in figure 11. The artificial databases are obtained in 10 trial runs of 1000 simulations with the second version of the rectangular-exchange method. On the right, the graph represents the number of different arrays in the simulations, for each trial run, and for each number of stages. As the number of stages increases more arrays are obtained, and the curve goes up. This ascension is similar for each trial run. There are about 1800 different artificial databases, which are M-equivalent to the original database, and with 1000 simulations and a choice of more than 28 stages, more than 900 artificial databases are obtained. This result is very satisfying.

The acceptance-rejection method consists in randomly permutating each column values, and conserving only the arrays whose row sums are equal to those of the original data. By a generation of 10 sets of 3000 arrays from the original array (fig. 10, left) 1855, 1812, 1866, 1861, 1882, 1879, 1828, 1882, 1867, 1867 M-equivalent arrays are successively obtained, evenly distributed (about 370) in the expected 5 different arrays. By an identical generation for another slightly greater array (fig. 11, left), between 53 and 76 different M-equivalent arrays are obtained, against the expected 1800. In a last trial run with a Boolean array of 33 rows and 6 columns, one M-equivalent array was obtained after about 40.000 non-M-equivalent artificial arrays were rejected. So the acceptance-rejection method is not efficient for finding one hundred simulated databases or more. With the rectangular-exchange method, each simulation produces exactly one M-equivalent artificial database. If the simulations with arrays that have many M-equivalent arrays are repeated, a great enough variety of these arrays is obtained to test the hypothesis of independence of database properties given the database structure. This latter method is by far more efficient than the acceptance-rejection one. With this method potential arrays can be found

faster, because to create an artificial array from an original array of n rows and p columns, a minimum of n' (n' can be less than n) times p exchanges of values for the acceptance-rejection may be operated, against 4 times the number of rectangular-exchanges for the rectangular-exchange method. Even if the time to decide if rectangular-exchange is possible is added, and also the time to determine the ideal number of rectangular-exchanges, the comparison is clearly to the advantage of the latter method. Moreover all the potential artificial databases being conserved, there is no reason to check them by the calculation of their row sums, contrary to the former method.

But instead of the acceptance-rejection method, the rectangular-exchange method is susceptible to be biased because the bias absence has not been proved. And not finding any bias for numerous little examples as has been done (only one, with equal expected frequencies, is presented in this paper) is not the proof of their absence, only a favorable impression. But for greater examples, the expected frequencies are generally equal to a value strictly between 0 and 1, and the obtained frequencies are generally 0 (absence) or rarely 1 (presence), so there is no point in checking for a bias (a difference between the expected frequency and the observed one which remains constant when simulations are repeated). In addition, for databases whose structures are so rigid that some frequencies higher than 1 are expected, some possible adaptations of the rectangular-exchange method have been investigated. So, the rectangular-exchange simulation method produces easily a set of one hundred or more simulated databases and they seem representative enough of all the possible databases, which are M-equivalent to the original databases, to be used in tests. Except perhaps for a very small number of particular databases which are not the present preoccupation in data mining, the rectangular-exchange simulation method largely surpasses the acceptance-rejection method in efficiency.

7. Conclusion and perspectives

The purpose of this paper has been to reinforce the quality of the set of the ARs extracted from a database. With common AR extraction, all the links present in the database are extracted, and since only a part of these links contribute to knowledge, researchers have defined numerous quality measures to help the pruning of ARs sets. But use of this pruning method has been shown to be a little hazardous. The proposed method is a statistical test which allows to keep only the robust itemsets, by deleting those corresponding to accidental links, due to chance. A randomization test has been chosen, which is more adapted to various databases than the other statistical tests, and the associated simulation method has been built to generate artificial databases similar to the original one except for the links between properties, which are deleted. With a first trial run of this test on a real database, it has been shown that particular links between the database properties, such as the links created by the probability law of its row sums, are found to be non-accidental. As the corresponding ARs seem irrelevant in an ARs set, the simulation method has been modified to delete them from the links which are tested. This last version of the test is a powerful tool for bringing out only the “context-free” links from a database and far more efficient than the acceptance- rejection randomization test.

But its powerfulness for itemsets with k number properties, where k is greater than 2, could be increased if these itemsets were considered as links between itemsets of k-1 properties instead of only links of k properties. A construction of this supplementary option is in progress.

This test is an itemset support test, but not precisely a test for a quality measure of ARs. ARs are created from itemsets, by pruning itemsets ARs are pruned. We could again prune ARs by a further test of this type, such as a confidence test.

And this test only works for Boolean properties. It would be interesting to extend it to quantitative ARs, such as fuzzy association rules (Delgado et al. 2003, Cadot et al. 2004).

Bibliography

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I. Press (1996). Fast discovery of association rules. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., eds., *Advances in Knowledge Discovery and Data Mining*. Menlo Park, California : AAAI Press , MIT Press. pp. 307-328.

Agresti, A. (1992) A survey of exact inference for contingency tables. Statistical Science (7):131-53

Bastide, R Y., Taouil, R., Pasquie,r N., Stumme, G. and Lakhal, L. (2002). Pascal : un algorithme d'extraction des motifs fréquents" , *Technique et science informatiques*, 21(1) :65-75.

Botta, M., Boulicaut, J.-F., Masson, C. and Meo, R. (2002). A Comparison between Query Languages for the Extraction of Association Rules. *Proceedings of DaWaK: Springer-Verlag LNCS volume 2454.* pp. 1-10

Breslau, L., Cao, P., Fan, L., Phillips, G. and Shenker S. (1999). Web Caching and Zipf-like Distributions : Evidence and Implications. *Proceedings of IEEE Infocom*, NewYork: pp. 126-134.

Briand, H., Sebag, M., Gras, R. and Guillet, F., ed. (2004). *Mesures de qualité pour la Fouille de Données* Toulouse, France : RNTI-E-1, Cepadues Editions.

Cadot, M., Maj, J.-B., Ziadé T., (2005) Association Rules and Statistics, dans *Encyclopedia of Data Warehousing and Mining,* John Wang ed. Montclair State University, USA, p. 74-77

Cadot, M. and Napoli, A. (2004) Règles d'association et codage flou des données. *11èmes Rencontres de la Société Francophone de Classification - SFC'04.* (Bordeaux) p130-133.

Delgado, M., Marin, N., Sanchez, D. and Vila, M.-A. (2003). Fuzzy association rules: general model and applications. *IEEE Transactions on fuzzy systems*, 11(2): 214- 225 ...

Edgington, E.S. (1995), *Randomization tests*, New York, USA: Marcel Decker

Gras R. (1979). *Contribution à l'étude expérimentale et à l'analyse de certaines acquisitions cognitives et de certains objectifs didactiques en mathématiques* , Thesis of Université de RennesI, France.

Guigues, J.L. and Duquenne, V. (1986) Familles minimales d'implications informatives résultant d'un tableau de données binaires, Math. Sci. Hum. (95). pp. 5-18

Guillet, F. (2004). Mesure de qualité des connaissances en ECD. *Tutorial of EGC 2004,* Clermont-ferrand, France.

Guiraud, P. (1960). *Problèmes et méthodes de la statistique linguistique*. Paris, France: Presses Universitaires de France.

Han, J. and Kamber, M., (2001). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers..

Hand, D., Mannila H. and Smyth P. (2001), *Principles of Data Mining*, Cambridge, Massachussets, USA: The MIT Press.

Hollander, M. and Wolfe, D.A. (1973). *Nonparametric statistical Methods*, New York: John Wiley & Sons, Inc.

Howel, D.C. (1997). *Statistical Methods for Psychology*, Duxbury: A Division of International Thomson Publishing Inc.

Legendre, P., Legendre, L. (1998). *Numerical Ecology*. Amsterdam, The Netherlands: Elsevier.

Manly, B.F.J. (1997) Randomization, *Bootstrap and Monte Carlo methods in Biology*, Boca Raton, Florida, USA: Chapman & Hall/CRC.

Snedecor et al., G. W., and Coochran, W. G. (1967). *Statistical methods*. 6th ed. Ames, IA: Iowa State University.

Winer, B.J., Brown D.R., and Michels, K.M. (1991). Statistical principles in experimental design 3rd ed. New York: McGraw-Hill.