Annales Mathematicae et Informaticae
***51** (2020) pp. 65–76*
DOI: 10.33039/ami.2020.07.009
https://ami.uni-eszterhazy.hu

Closed Association Rules

Laszlo Szathmary

University of Debrecen, Faculty of Informatics
Debrecen, Hungary
szathmary.laszlo@inf.unideb.hu

Submitted: February 4, 2020
Accepted: July 10, 2020
Published online: July 23, 2020

Abstract

In this paper we present a new basis for association rules called Closed Association Rules ($\mathcal{CR}$). This basis contains all valid association rules that can be generated from frequent closed itemsets. $\mathcal{CR}$ is a lossless representation of all association rules. Regarding the number of rules, our basis is between all association rules ($\mathcal{AR}$) and minimal non-redundant association rules ($\mathcal{MNR}$), filling a gap between them. The new basis provides a framework for some other bases and we show that $\mathcal{MNR}$ is a subset of $\mathcal{CR}$. Our experiments show that $\mathcal{CR}$ is a good alternative for all association rules. The number of generated rules can be much less, and beside frequent closed itemsets nothing else is required.

1. Introduction

In data mining, frequent itemsets (FIs) and association rules play an important role [2]. Generating valid association rules (denoted by $\mathcal{AR}$) from frequent itemsets often results in a huge number of rules, which limits their usefulness in real life applications. To solve this problem, different concise representations of association rules have been proposed, e.g. generic basis ($\mathcal{GB}$), informative basis ($\mathcal{IB}$) [3], Duquennes-Guigues basis ($\mathcal{DG}$) [5], Luxenburger basis ($\mathcal{LB}$) [8], etc. A very good comparative study of these bases can be found in [7], where it is stated that a rule representation should be *lossless* (should enable the derivation of all valid rules), *sound* (should forbid the derivation of rules that are not valid) and *informative* (should allow the determination of rules parameters such as support and confidence).

In this paper we present a new basis for association rules called Closed Association Rules ($\mathcal{CR}$). The number of rules in $\mathcal{CR}$ is less than the number of all rules, especially in the case of dense, highly correlated data when the number of frequent itemsets is much more than the number of frequent closed itemsets. $\mathcal{CR}$ contains more rules than minimal non-redundant association rules ($\mathcal{MNR}$), but for the extraction of closed association rules we *only* need frequent closed itemsets, nothing else. On the contrary, the extraction of $\mathcal{MNR}$ needs much more computation since frequent generators also have to be extracted and assigned to their closures.[1]

The remainder of the paper is organized as follows. Background on pattern mining and concept analysis is provided in Section 2. All association rules, closed association rules and minimal non-redundant association rules are presented in Sections 3, 4 and 5, respectively. Experimental results are provided in Section 6, and Section 7 concludes the paper.

2. Basic concepts

In the following, we recall basic concepts from frequent pattern mining and formal concept analysis (FCA). The following 5×5 sample dataset: $\mathcal{D} = \{(1,\ ABDE),\ (2,\ AC),\ (3,\ ABCE),\ (4,\ BCE)\},\ (5,\ ABCE)\}$ will be used as a running example. Henceforth, we refer to it as dataset $\boldsymbol{\mathcal{D}}$.

Frequent itemsets. We consider a set of *objects* $O = \{o_1, o_2, \ldots, o_m\}$, a set of *attributes* $A = \{a_1, a_2, \ldots, a_n\}$, and a binary relation $R \subseteq O \times A$, where $R(o, a)$ means that the object o has the attribute a. In formal concept analysis the triple (O, A, R) is called a *formal context* [4]. The Galois connection for (O, A, R) is defined along the lines of [4] in the following way (here $B \subseteq O$, $D \subseteq A$):

$$B' = \{a \in A \mid R(o, a) \text{ for all } o \in B\}, \quad D' = \{o \in O \mid R(o, a) \text{ for all } a \in D\}.$$

In data mining applications, an element of A is called an *item* and a subset of A is called an *itemset*. Further on, we shall keep to these terms. An itemset of size i is called an i-itemset.[2] We say that an itemset $P \subseteq A$ *belongs* to an object $o \in O$, if $(o, p) \in R$ for all $p \in P$, or $P \subseteq o'$. The *support* of an itemset $P \subseteq A$ indicates the number of objects to which the itemset belongs: $supp(P) = |P'|$. An itemset is *frequent* if its support is not less than a given *minimum support* (denoted by *min_supp*). An itemset P is *closed* if there exists no proper superset with the same support. The closure of an itemset P (denoted by P'') is the largest superset of P with the same support. Naturally, if $P = P''$, then P is a closed itemset. The task of frequent itemset mining consists of generating all (closed) itemsets (with their supports) with supports greater than or equal to a specified *min_supp*.

Two itemsets $P, Q \subseteq A$ are said to be *equivalent* ($P \cong Q$) iff they belong to the same set of objects (i.e. $P' = Q'$). The set of itemsets that are equivalent to

[1] Concepts in this section are defined in Section 2.

[2] For instance, $\{A, B, E\}$ is a 3-itemset. Further on we use separator-free set notations, i.e. ABE stands for $\{A, B, E\}$.

an itemset P (P's *equivalence class*) is denoted by $[P] = \{Q \subseteq A \mid P \cong Q\}$. An itemset $P \in [P]$ is called a *generator*, if P has no proper subset in $[P]$, i.e. it has no proper subset with the same support. A *frequent generator* is a generator whose support is not less than a given minimum support.

Frequent association rules. An association rule is an expression of the form $P_1 \to P_2$, where P_1 and P_2 are arbitrary itemsets ($P_1, P_2 \subseteq A$), $P_1 \cap P_2 = \emptyset$ and $P_2 \neq \emptyset$. The left side, P_1 is called *antecedent*, the right side, P_2 is called *consequent*. The (absolute) support of an association rule r is defined as: $supp(r) = supp(P_1 \cup P_2)$. The *confidence* of an association rule $r\colon P_1 \to P_2$ is defined as the conditional probability that an object has itemset P_2, given that it has itemset P_1: $conf(r) = supp(P_1 \cup P_2)/supp(P_1)$. An association rule is *valid* if $supp(r) \geq min\_supp$ and $conf(r) \geq min\_conf$. The set of all valid association rules is denoted by $\mathcal{AR}$.

Minimal non-redundant association rules ($\mathcal{MNR}$) [3] have the following form: $P \to Q \setminus P$, where $P \subset Q$, P is a generator and Q is a closed itemset. That is, an $\mathcal{MNR}$ rule has a minimal antecedent and a maximal consequent. Minimal (resp. maximal) means that the antecedent (resp. consequent) is a minimal (resp. maximal) element in its equivalence class. Note that P and Q are not necessarily in the same equivalence class. As it was shown in [3], $\mathcal{MNR}$ rules contain the most information among rules with the same support and same confidence.

3. All association rules

From now on, by "all association rules" we mean all (frequent) *valid* association rules. The concept of association rules was introduced by Agrawal *et al.* [1]. Originally, the extraction of association rules was used on sparse market basket data. The first efficient algorithm for this task was *Apriori*. The generation of all valid association rules consists of two main steps:

1. Find all *frequent* itemsets P in a dataset, i.e. where $supp(P) \geq min\_supp$.
2. For each frequent itemset P_1 found, generate all confident association rules r of the form $P_2 \to (P_1 \setminus P_2)$, where $P_2 \subset P_1$ and $conf(r) \geq min\_conf$.

The more difficult task is the first step, which is computationally and I/O intensive.

Generating all valid association rules. Once all frequent itemsets and their supports are known, this step can be done in a relatively straightforward manner. The general idea is the following: for every frequent itemset P_1, all subsets P_2 of P_1 are derived, and the ratio $supp(P_1)/supp(P_2)$ is computed.[3] If the result is higher or equal to min_conf, then the rule $P_2 \to (P_1 \setminus P_2)$ is generated.

[3] $supp(P_1)/supp(P_2)$ is the confidence of the rule $P_2 \to (P_1 \setminus P_2)$.

The support of any subset P_3 of P_2 is greater than or equal to the support of P_2. Thus, the confidence of the rule $P_3 \rightarrow (P_1 \setminus P_3)$ is necessarily less than or equal to the confidence of the rule $P_2 \rightarrow (P_1 \setminus P_2)$. Hence, if the rule $P_2 \rightarrow (P_1 \setminus P_2)$ is not confident, then neither is the rule $P_3 \rightarrow (P_1 \setminus P_3)$. Conversely, if the rule $(P_1 \setminus P_2) \rightarrow P_2$ is confident, then all rules of the form $(P_1 \setminus P_3) \rightarrow P_3$ are confident. For example, if the rule $A \rightarrow BE$ is confident, then the rules $AB \rightarrow E$ and $AE \rightarrow B$ are confident as well.

Using this property for efficiently generating valid association rules, the algorithm works as follows [1]. For each frequent itemset P_1, all *confident* rules with one item in the consequent are generated. Then, using the `Apriori-Gen` function (from [1]) on the set of 1-long consequents, we generate consequents with 2 items. Only those rules with 2 items in the consequent are kept whose confidence is greater than or equal to min_conf. The 2-long consequents of the confident rules are used for generating consequents with 3 items, etc.

Example. Table 1 depicts which valid association rules ($\mathcal{AR}$) can be extracted from dataset $\mathcal{D}$ with $min\_supp = 3$ (60%) and $min\_conf = 0.5$ (50%). First, all frequent itemsets have to be extracted from the dataset. In $\mathcal{D}$ with $min\_supp = 3$ there are 12 frequent itemsets, namely A (supp: 4), B (4), C (4), E (4), AB (3), AC (3), AE (3), BC (3), BE (4), CE (3), ABE (3) and BCE (3).[4] Only those itemsets can be used for generating association rules that contain at least 2 items. Eight itemsets satisfy this condition. For instance, using the itemset ABE, which is composed of 3 items, the following rules can be generated: $BE \rightarrow A$ (supp: 3; conf: 0.75), $AE \Rightarrow B$ (3; 1.0) and $AB \Rightarrow E$ (3; 1.0). Since all these rules are confident, their consequents are used to generate 2-long consequents: AB, AE and BE. This way, the following rules can be constructed: $E \rightarrow AB$ (3; 0.75), $B \rightarrow AE$ (3; 0.75) and $A \rightarrow BE$ (3; 0.75). In general, it can be said that from an m-long itemset, one can potentially generate $2^m - 2$ association rules.

4. Closed Association Rules

In the previous section we presented all association rules that are generated from frequent itemsets. Unfortunately, the number of these rules can be very large, and many of these rules are redundant, which limits their usefulness. Applying concise rule representations (a.k.a. *bases*) with appropriate inference mechanisms can lessen the problem [7]. By definition, a *concise representation of association rules* is a subset of all association rules with the following properties: **(1)** it is much smaller than the set of all association rules, and **(2)** the whole set of all association rules can be restored from this subset (possibly with no access to the database, i.e. very efficiently) [6].

[4] Support values are indicated in parentheses.

$\mathcal{AR}$	supp.	conf.	$\mathcal{CR}$	$\mathcal{MNR}$
$B \to A$	3	0.75		
$A \to B$	3	0.75		
$C \to A$	3	0.75	+	+
$A \to C$	3	0.75	+	+
$E \to A$	3	0.75		
$A \to E$	3	0.75		
$C \to B$	3	0.75		
$B \to C$	3	0.75		
$E \Rightarrow B$	4	1.0	+	+
$B \Rightarrow E$	4	1.0	+	+
$E \to C$	3	0.75		
$C \to E$	3	0.75		
$BE \to A$	3	0.75	+	
$AE \Rightarrow B$	3	1.0	+	+
$AB \Rightarrow E$	3	1.0	+	+
$E \to AB$	3	0.75	+	+
$B \to AE$	3	0.75	+	+
$A \to BE$	3	0.75	+	+
$CE \Rightarrow B$	3	1.0	+	+
$BE \to C$	3	0.75	+	
$BC \Rightarrow E$	3	1.0	+	+
$E \to BC$	3	0.75	+	+
$C \to BE$	3	0.75	+	+
$B \to CE$	3	0.75	+	+

Table 1: Different sets of association rules extracted from dataset $\mathcal{D}$ with $min\_supp = 3$ (60%) and $min\_conf = 0.5$ (50%)

Related work. In addition to the first method presented in the previous section, there is another approach for finding all association rules. This approach was introduced in [9] by Bastide *et al.* They have shown that frequent closed itemsets are a lossless, condensed representation of frequent itemsets, since the whole set of frequent itemsets can be restored from them with the proper support values. They propose the following method for finding all association rules. First, they extract frequent closed itemsets[5], then they restore the set of frequent itemsets from them, and finally they generate all association rules. The number of FCIs is usually much less than the number of FIs, especially in dense and highly correlated datasets. In such databases the exploration of all association rules can be done more efficiently by this way. However, this method has some disadvantages: **(1)** the restoration of FIs from FCIs needs *lots of* memory, **(2)** the final result is still "all the association rules", which means lots of redundant rules.

[5]For this task they introduced a new algorithm called "Close". Close is a levelwise algorithm for finding FCIs.

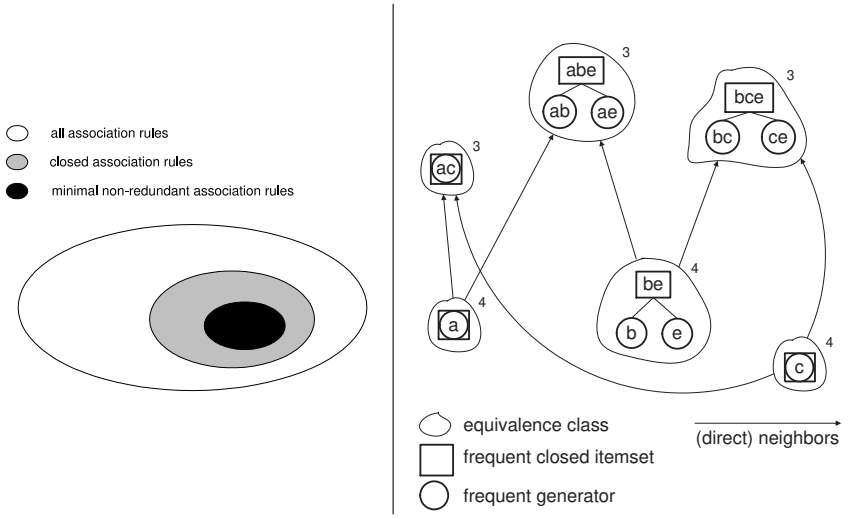

Figure 1: **Left:** position of Closed Rules; **Right:** equivalence classes of $\mathcal{D}$ with $min\_supp = 3$ (60%). Support values are indicated in the top right corners.

Contribution. We introduce a new basis called Closed Association Rules, or simply Closed Rules ($\mathcal{CR}$). This basis requires frequent closed itemsets only. The difference between our work and the work presented in [9] stems from the fact that although we also extract FCIs, instead of restoring all FIs from them, we use them *directly* to generate valid association rules. This way, we find less and probably more interesting association rules.

$\mathcal{CR}$ is a generating set for all valid association rules with their proper support and confidence values. Our basis fills a gap between all association rules and minimal non-redundant association rules ($\mathcal{MNR}$), as depicted in Figure 1 (left). $\mathcal{CR}$ contains all valid rules that are derived from frequent closed itemsets. Since the number of FCIs are usually much less than the number of FIs, the number of rules in our basis is also much less than the number of all association rules. Using our basis the restoration of all valid association rules can be done without any loss of information. It is possible to deduce efficiently, without access to the dataset, all valid association rules with their supports and confidences from this basis, since frequent closed itemsets are a lossless representation of frequent itemsets. Furthermore, we will show in the next section that minimal non-redundant association rules are a special subset of the Closed Rules, i.e. $\mathcal{MNR}$ can be defined in the framework of our basis. $\mathcal{CR}$ has the advantage that its rules can be generated very easily since only the frequent closed itemsets are needed. As there are usually much less FCIs than FIs, the derivation of the Closed Rules can be done much more efficiently than generating all association rules.

Before showing our algorithm for finding the Closed Rules, we present the essential definitions.

Definition 4.1 (closed association rule)**.** An association rule $r\colon P_1 \to P_2$ is called *closed* if $P_1 \cup P_2$ is a closed itemset.

This definition means that the rule is derived from a closed itemset.

Definition 4.2 (Closed Rules)**.** Let FC be the set of frequent closed itemsets. The set of Closed Rules contains *all* valid closed association rules:

$$\mathcal{CR} = \{r\colon P_1 \to P_2 \mid (P_1 \cup P_2) \in FC \wedge supp(r) \geq min\_supp \wedge conf(r) \geq min\_conf\}.$$

Property 4.3. *The support of an arbitrary frequent itemset is equal to the support of its smallest frequent closed superset [9].*

By this property, FCIs are a condensed lossless representation of FIs. This is also called the *frequent closed itemset representation* of frequent itemsets. Property 4.3 can be generalized the following way:

Property 4.4. *If an arbitrary itemset X has a frequent closed superset, then X is frequent and its support is equal to the support of its smallest frequent closed superset. If X has no frequent closed superset, then X is not frequent.*

The algorithm. The idea behind generating all valid association rules is the following. First we need to extract all frequent itemsets. Then rules of the form $X \setminus Y \to Y$, where $Y \subset X$, are generated for all frequent itemsets X, provided the rules have at least minimum confidence.

Finding closed association rules is done similarly. However, this time we only have frequent *closed* itemsets available. In this case the left side of a rule $X \setminus Y$ can be non-closed. For calculating the confidence of rules its support must be known. Thanks to Property 4.3, this support value can be calculated by only using frequent closed itemsets. It means that only FCIs are needed; all frequent itemsets do not have to be extracted. This is the principle idea behind this part of our work.

Example. Table 1 depicts which closed association rules ($\mathcal{CR}$) can be extracted from dataset $\mathcal{D}$ with $min\_supp = 3$ (60%) and $min\_conf = 0.5$ (50%). First, frequent closed itemsets must be extracted from the dataset. In $\mathcal{D}$ with $min\_supp = 3$ there are 6 FCIs, namely A (supp: 4), C (4), AC (3), BE (4), ABE (3) and BCE (3). Note that the total number of frequent itemsets by these parameters is 12. Only those itemsets can be used for generating association rules that contain at least 2 items. There are 4 itemsets that satisfy this condition, namely itemsets AC (supp: 3), BE (4), ABE (3) and BCE (3). Let us see which rules can be generated from the itemset BCE for instance. Applying the algorithm from [1], we get three rules: $CE \to B$, $BE \to C$ and $BC \to E$. Their support is known, it is equal to the support of BCE. To calculate the confidence values we need to know the support of the left sides too. The support of BE is known since it is a closed

itemset, but CE and BC are non-closed. Their supports can be derived by Property 4.3. The smallest frequent closed superset of both CE and BC is BCE, thus their supports are equal to the support of this closed itemset, which is 3. Then, using the algorithm from [1], we can produce three more rules: $E \rightarrow BC$, $C \rightarrow BE$ and $B \rightarrow CE$. Their confidence values are calculated similarly. From the four frequent closed itemsets 16 closed association rules can be extracted altogether, as depicted in Table 1.

5. Minimal non-redundant association rules

As seen in Section 2, minimal non-redundant association rules ($\mathcal{MNR}$) have the following form: $P \rightarrow Q \setminus P$, where $P \subset Q$, P is a generator and Q is a closed itemset.

In order to generate these rules efficiently, one needs to extract the frequent closed itemsets (FCIs), the frequent generators (FGs), and then these itemsets must be grouped together. That is, to generate these rules, one needs to explore all the frequent equivalence classes in a dataset (see Figure 1, right). Most algorithms address either FCIs or FGs, and only few algorithms can extract both types of itemsets.

Example. Table 1 depicts which $\mathcal{MNR}$ rules can be extracted from dataset $\mathcal{D}$ with $min\_supp = 3$ (60%) and $min\_conf = 0.5$ (50%). As can be seen, there are 14 $\mathcal{MNR}$ rules in the dataset. For instance, $BE \rightarrow A$ is not an $\mathcal{MNR}$ rule becasue its antecedent (BE) is not a generator (see Figure 1, right). To learn more about the $\mathcal{MNR}$ rules, please refer to [11].

Comparing $\mathcal{CR}$ and $\mathcal{MNR}$. As we have seen, $\mathcal{CR}$ is a maximal set of closed association rules, i.e. it contains *all* closed association rules. As a consequence, we cannot say that this basis is minimal, or non-redundant, but by all means it is a smaller set than $\mathcal{AR}$, especially in the case of dense, highly correlated datasets. Moreover, $\mathcal{CR}$ is a framework for some other bases. For instance, minimal non-redundant association rules are also closed association rules, since by definition the union of the antecedent and the consequent of such a rule forms a frequent closed itemset. Thus, $\mathcal{MNR}$ is a special subset of $\mathcal{CR}$, which could also be defined the following way:

Definition 5.1. Let CR be the set of Closed Rules. The set of minimal non-redundant association rules is:

$$\mathcal{MNR} = \{r\colon P_1 \rightarrow P_2 \mid r \in CR \wedge P_1 \text{ is a frequent generator}\}.$$

This is equivalent to the following definition:

$$\mathcal{MNR} = \{r\colon P_1 \rightarrow P_2 \mid (P_1 \cup P_2) \in FC \wedge P_1 \text{ is a frequent generator}\},$$

where FC stands for the set of frequent closed itemsets.

6. Experimental results

For comparing the different sets of association rules ($\mathcal{AR}$, $\mathcal{CR}$ and $\mathcal{MNR}$), we used the multifunctional *Zart* algorithm [11] from the CORON[6] system [10]. *Zart* was implemented in Java. The experiments were carried out on an Intel Pentium IV 2.4 GHz machine running Debian GNU/Linux with 2 GB RAM. All times reported are real, wall clock times as obtained from the Unix *time* command between input and output. For the experiments we have used the following datasets: T20I6D100K, C20D10K and MUSHROOMS.[7] It has to be noted that T20 is a sparse, weakly correlated dataset imitating market basket data, while the other two datasets are dense and highly correlated. Weakly correlated data usually contain few frequent itemsets, even at low minimum support values, and almost all frequent itemsets are closed. On the contrary, in the case of highly correlated data the difference between the number of frequent itemsets and frequent closed itemsets is significant.

6.1. Number of rules

Table 2 shows the following information: minimum support and confidence; number of all association rules; number of closed rules; number of minimal non-redundant association rules. We attempted to choose significant min_supp and min_conf thresholds as observed in other papers for similar experiments.

In T20 almost all frequent itemsets are closed, thus the number of all rules and the number of closed association rules is almost equal. For the other two datasets that are dense and highly correlated, the reduction of the number of rules in the Closed Rules is considerable.

The size of the $\mathcal{MNR}$ set is almost equal to the size of $\mathcal{AR}$ in sparse datasets, but in dense datasets $\mathcal{MNR}$ produces much less rules.

6.2. Execution times of rule generation

Figure 3 shows for each dataset the execution times of the computation of all, closed and minimal non-redundant association rules. For the extraction of the necessary itemsets we used the multifunctional *Zart* algorithm [11] that can generate all three kinds of association rules. Figure 3 does not include the extraction time of itemsets, it only shows the time of rule generation.

For datasets with much less frequent closed itemsets (C20, MUSHROOMS), the generation of closed rules is more efficient than finding all association rules. As seen before, we need to look up the closed supersets of frequent itemsets very often when extracting closed rules. For this procedure we use the trie data structure that shows its advantage on dense, highly correlated datasets. On the contrary, when almost all frequent itemsets are closed (T20), the high number of superset operations cause that all association rules can be extracted faster.

[6] `http://coron.loria.fr`

[7] `https://github.com/jabbalaci/Talky-G/tree/master/datasets`

dataset (min_supp)	min_conf	$\mathcal{AR}$	$\mathcal{CR}$	$\mathcal{MNR}$
$\mathcal{D}$ (40%)	50%	50	30	25
T20I6D100K (0.5%)	90%	752,715	726,459	721,948
	70%	986,058	956,083	951,572
	50%	1,076,555	1,044,086	1,039,575
	30%	1,107,258	1,073,114	1,068,603
C20D10K (30%)	90%	140,651	47,289	9,221
	70%	248,105	91,953	19,866
	50%	297,741	114,245	25,525
	30%	386,252	138,750	31,775
Mushrooms (30%)	90%	20,453	5,571	1,496
	70%	45,147	11,709	3,505
	50%	64,179	16,306	5,226
	30%	78,888	21,120	7,115

Table 2: Comparing sizes of different sets of association rules

dataset (min_supp)	min_conf	$\mathcal{AR}$	$\mathcal{CR}$	$\mathcal{MNR}$
T20I6D100K (0.5%)	90%	114.43	120.30	394.14
	70%	147.69	152.31	428.59
	50%	165.48	167.07	441.52
	30%	169.66	170.06	449.47
C20D10K (30%)	90%	15.72	12.49	1.68
	70%	26.98	21.10	2.77
	50%	34.74	24.24	3.35
	30%	41.40	27.36	4.04
Mushrooms (30%)	90%	1.93	1.49	0.54
	70%	3.99	2.44	0.78
	50%	5.63	2.98	1.00
	30%	6.75	3.31	1.28

Table 3: Execution times of rule generation (given is seconds)

Experimental results show that $\mathcal{CR}$ can be generated more efficiently than $\mathcal{MNR}$ on sparse datasets. However, on dense datasets $\mathcal{MNR}$ can be extracted much more efficiently.

7. Conclusion

In this paper we presented a new basis for association rules called Closed Rules ($\mathcal{CR}$). This basis contains all valid association rules that can be generated from frequent closed itemsets. $\mathcal{CR}$ is a lossless representation of all association rules. Regarding the number of rules, our basis is between all association rules ($\mathcal{AR}$) and minimal non-redundant association rules ($\mathcal{MNR}$), filling a gap between them. The

new basis provides a framework for some other bases. We have shown that $\mathcal{MNR}$ is a subset of $\mathcal{CR}$. The number of extracted rules is less than the number of all rules, especially in the case of dense, highly correlated data when the number of frequent itemsets is much more than the number of frequent closed itemsets. $\mathcal{CR}$ contains more rules than $\mathcal{MNR}$, but for the extraction of closed association rules we *only* need frequent closed itemsets, nothing else. On the contrary, the extraction of minimal non-redundant association rules needs much more computation since frequent generators also have to be extracted and assigned to their closures.

As a summary, we can say that $\mathcal{CR}$ is a good alternative for all association rules. The number of generated rules can be much less, and beside frequent closed itemsets nothing else is required.

Acknowledgement

This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo: *Fast discovery of association rules*, in: Advances in knowledge discovery and data mining, American Association for Artificial Intelligence, 1996, pp. 307–328, isbn: 0-262-56097-6.

[2] R. Agrawal, R. Srikant: *Fast Algorithms for Mining Association Rules in Large Databases*, in: Proc. of the 20th Intl. Conf. on Very Large Data Bases (VLDB '94), San Francisco, CA: Morgan Kaufmann, 1994, pp. 487–499, isbn: 1-55860-153-8.

[3] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, L. Lakhal: *Mining Minimal Non-Redundant Association Rules Using Frequent Closed Itemsets*, in: Proc. of the Computational Logic (CL '00), vol. 1861, LNAI, Springer, 2000, pp. 972–986.

[4] B. Ganter, R. Wille: *Formal concept analysis: mathematical foundations*, Berlin / Heidelberg: Springer, 1999, p. 284, isbn: 3540627715.

[5] J. L. Guigues, V. Duquenne: *Familles minimales d'implications informatives résultant d'un tableau de données binaires*, Mathématiques et Sciences Humaines 95 (1986), pp. 5–18.

[6] B. Jeudy, J.-F. Boulicaut: *Using condensed representations for interactive association rule mining*, in: Proc. of PKDD '02, volume 2431 of LNAI, Helsinki, Finland, Springer-Verlag, 2002, pp. 225–236.

[7] M. Kryszkiewicz: *Concise Representations of Association Rules*, in: Proc. of the ESF Exploratory Workshop on Pattern Detection and Discovery, 2002, pp. 92–109.

[8] M. Luxenburger: *Implications partielles dans un contexte*, Mathématiques, Informatique et Sciences Humaines 113 (1991), pp. 35–55.

[9] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal: *Efficient mining of association rules using closed itemset lattices*, Inf. Syst. 24.1 (1999), pp. 25–46, issn: 0306-4379, doi: http://dx.doi.org/10.1016/S0306-4379(99)00003-4.

[10] L. Szathmary: *Symbolic Data Mining Methods with the Coron Platform*, PhD Thesis in Computer Science, Univ. Henri Poincaré – Nancy 1, France, Nov. 2006.

[11] L. Szathmary, A. Napoli, S. O. Kuznetsov: *ZART: A Multifunctional Itemset Mining Algorithm*, in: Proc. of the 5th Intl. Conf. on Concept Lattices and Their Applications (CLA '07), Montpellier, France, Oct. 2007, pp. 26–37,
url: `http://hal.inria.fr/inria-00189423/en/`.