Meta-learning for Post-processing of Association Rules

Petr Berka[1,2] and Jan Rauch[1]

[1] University of Economics, W. Churchill Sq. 4, 130 67 Prague
[2] Institute of Finance and Administration, Estonska 500, 101 00 Prague

Abstract. The paper presents a novel approach to post-processing of association rules based on the idea of meta-learning. A subsequent association rule mining step is applied to the results of "standard" association rule mining. We thus obtain "rules about rules" that help to better understand the association rules generated in the first step.

We define various types of such meta-rules and report some experiments on UCI data. When evaluating the proposed method, we use the *apriori* algorithm implemented in Weka.

1 Introduction

The term association rules was coined by R. Agrawal in the early 90th in relation to so called market basket analysis [2]. In this analysis, transaction data recorded by point-of-sale (POS) systems in supermarkets are analyzed in order to understand the purchase behavior of groups of customers, and use it to increase sales, and for cross-selling, store design, discount plans and promotions. This idea of association rules has been later generalized to any data in the tabular, attribute-value form. So data describing properties (values of attributes) of some examples can be analyzed in order to find associations between conjunctions of attribute-value pairs (categories). Let us denote these conjunctions as *Ant* and *Suc* and the association rule as

$$Ant \Longrightarrow Suc.$$

The two basic characteristics of an association rule are *support* and *confidence*. Support is the estimate of the probability $P(Ant \wedge Suc)$, (the frequency of $Ant \wedge Suc$ is the *absolute support*), confidence is the estimate of the probability $P(Suc|Ant)$. So an example of a rule based on Table 1 is

$$income(high) \wedge balance(high) \Longrightarrow loan(yes)$$

with the support 0.1667 and the confidence 1.

In association rule discovery the task is to find all syntactically correct rules $Ant \Longrightarrow Suc$ (i.e. rules, in which two different values of an attribute cannot occur) such that the support and confidence of the rules are above the userdefined thresholds *minconf* and *minsup*. There is a number of algorithms, that perform

T.B. Pedersen, M.K. Mohania, and A M. Tjoa (Eds.): DaWaK 2010, LNCS 6263, pp. 251–262, 2010.

Table 1. Running example data

client	income	balance	sex	unemployed	loan
c1	high	high	female	no	yes
c2	high	high	male	no	yes
c3	low	low	male	no	no
c4	low	high	female	yes	yes
c5	low	high	male	yes	yes
c6	low	low	female	yes	no
c7	high	low	male	no	yes
c8	high	low	female	yes	yes
c9	low	medium	male	yes	no
c10	high	medium	female	no	yes
c11	low	medium	female	yes	no
c12	low	medium	male	no	yes

this task. The main idea of these algorithm is to repeatedly generate a rule in a "top-down" way by rule specialization (i.e. by adding categories to an existing combination) and test, if this rule meets the thresholds *minconf* and *minsup.* The probably best-known algorithm called *apriori* proceeds in two steps. All frequent itemsets are found in the first step. A frequent itemset is a set of items that is included in at least *minsup* transactions. Then, association rules with a confidence of at least *minconf* are generated in the second step [2].

There is also an alternative approach to association rules mining, the so called GUHA method that originates from the research of a group of Czech researchers from mid. 60th [6]. The aim of the GUHA method is to offer all interesting facts hidden in the analyzed data and relevant to the given problem. The method is realized by GUHA-procedures. The input of the GUHA procedure consists of the analyzed data and of a simple definition of a set of relevant (i.e. potentially interesting) patterns. GUHA procedure automatically generates each particular pattern and tests if it is true in the analyzed data. The output of the procedure consists of all prime patterns. The pattern is prime if it is true in the analyzed data and if it does not immediately follow from the other more simple output patterns [6].

The most important GUHA procedure is the procedure ASSOC [6]. This procedure mines for patterns that can be understood as a generalization of patterns now called association rules [2]. The most used current implementation of procedure ASSOC is the procedure *4ft-Miner* [9]. This procedure mines for association rules of the form

$$\varphi \approx \psi \quad \text{or} \quad \varphi \approx \psi/\chi$$

where φ, ψ, and χ are Boolean attributes. The rule $\varphi \approx \psi$ means that φ and ψ are associated in the way given by the symbol $\approx$. The conditional rule $\varphi \approx \psi/\chi$ means that φ and ψ are associated in the way given by the symbol $\approx$ if the condition given by χ is satisfied. The symbol $\approx$ is called *4ft-quantifier*, φ is called *antecedent*, ψ is called *succedent* and χ is *condition.* The *4ft-quantifier* $\approx$

corresponds to a condition concerning a four-fold contingency table $4ft(\varphi, \psi, \mathcal{M})$ of φ and ψ in the analyzed data matrix $\mathcal{M}$.

The Boolean attributes φ, ψ, and χ are automatically derived from the columns of the analyzed data matrix. *Basic Boolean attributes* are created first. The basic Boolean attribute has a form of $A(\alpha)$ where A is an attribute i.e. a column of the analyzed data matrix and α is a set of its possible values. The basic Boolean attribute $A(\alpha)$ is true in the row o of the data matrix if the value $A(o)$ of attribute A in row o belongs to α, formally if $A(o) \in \alpha$. An example of basic Boolean attribute of the data matrix in Tab. 1 is balance(medium, high) which is true for clients c1, c2, c4, c5, c9, c10, c11, c12 and false for clients c3, c6, c7, c8. *Literal* is basic Boolean attribute $A(\alpha)$ or its negation $\neg A(\alpha)$. *Partial cedent* is a conjunction or a disjunction of literals. Antecedent, succedent and condition are conjunctions of partial cedents. There are very fine tools to define a set of association rules to be generated and verified.

The rule $\varphi \approx \psi$ is true in analyzed data matrix $\mathcal{M}$ if the condition related to 4ft-quantifier $\approx$ is satisfied in four-fold contingency table $4ft(\varphi, \psi, \mathcal{M})$ of φ and ψ in data matrix $\mathcal{M}$. It is a quadruple $\langle a, b, c, d\rangle$ where a is the number of rows of $\mathcal{M}$ satisfying both φ and ψ, b is the number of rows of $\mathcal{M}$ satisfying φ and not satisfying ψ etc., see Tab. 2. The conditional association rule $\varphi \approx \psi/\chi$ is true in data matrix $\mathcal{M}$ if the rule $\varphi \approx \psi$ is true in data matrix $\mathcal{M}/\chi$. Data matrix $\mathcal{M}/\chi$ consists of all rules of $\mathcal{M}$ satisfying χ.

Table 2. $4ft(\varphi, \psi, \mathcal{M})$

$\mathcal{M}$	ψ	$\neg\psi$
φ	a	b
$\neg\varphi$	c	d

There are 17 various 4ft-quantifiers implemented in *4ft-Miner*. An example is the 4ft-quantifier $\sim^+_{p,B}$ of *above average dependence* which is defined in [8] by the condition $\frac{a}{a+b} \geq (1+p)\frac{a+c}{a+b+c+d} \wedge a \geq B$ for $0 < p$ and $B > 0$. This means that the relative frequency of objects satisfying ψ among the objects satisfying φ is at least $100p$ per cent higher than the relative frequency of objects satisfying ψ among all the observed objects and that there are at least B objects satisfying both φ and ψ. Thus an example of a rule found in data shown in Table 1 can be

$$balance(medium, high) \wedge \neg(unemployed(yes)) \sim^+_{0.5,100} loan(yes) \ .$$

This rule says that among clients with medium or high balance on account which are not unemployed, there are 50 % higher frequency of clients which will get the loan that among all clients and that there are at least 100 such clients.

So

- GUHA method offers more types of relations (so called quantifiers) between *Ant* and *Suc*,
- GUHA method offers more expressive syntax of *Ant* and *Suc*.

But still the algorithms for mining this type of rules are based on generating and testing of huge set of potential rules.

The main drawback of the association rules mining is the fact, that the result of an analysis will consist of many (hundreds, thousands) rules which have to be visually interpreted and evaluated by the domain expert. So some kind of post-processing of the results would be very helpful for the user. And indeed, various approaches have been used to post-process the huge list of found associations: filtering, selection, visualization, grouping and clustering. In our paper, we present an alternative approach based on the idea of meta-learning.

The rest of the paper is organized as follows: section 2 defines the association meta-rules, section 3 shows experimental evaluation of the proposed method, section 4 reviews other approaches to association rules post-processing and section 5 gives directions for our future work.

2 Association Meta-rules

The inspiration of our method comes from the area of meta-learning. Meta learning is a subfield of machine learning where automatic learning algorithms are applied to meta-data about machine learning experiments. The mostly used approaches to meta-learning (or combining classifiers) are bagging, boosting and stacking [4]. In bagging each classifier in the ensemble votes with equal weight when classifying new example; in order to promote model variance, bagging trains each classifier in the ensemble using a randomly-drawn subset of the training set. In boosting the ensemble of classifiers is built incrementally by training each new classifier to emphasize the training instances that previous classifiers miss-classified. In stacking a meta-classifier is build on top of the results of so called base classifiers that are each separately trained to classify the data.

We propose to apply association rule mining algorithm to the set of original association rules obtained as a result of a particular data mining task. This idea thus follows the stacking concept that is used to combine classifiers, but that has not been presented yet for descriptive tasks. The input to the proposed meta-learning step will be association rules encoded in a way suitable for association rule mining algorithm; the result will be a set of association meta-rules uncovering relations between various characteristics of the original set of rules.

We will distinguish two types of association meta-rules: *qualitative* and *quantitative*. Qualitative rules will represent the meta-knowledge in the form "if original association rules contain a conjunction of categories Ant, then they also contain the conjunction of categories Suc", i.e qualitative rules have the form

$$Ant \Longrightarrow Suc.$$

Quantitative rules will represent the meta-knowledge in the form "if original association rules contain a conjunction of categories Ant, then they have quantitative characteristics Q", i.e

$$Ant \Longrightarrow Q.$$

or, "if original association rules have quantitative characteristics Q, then they contain a conjunction of categories *Suc*", i.e

$$Q \Longrightarrow Suc.$$

where Q can be e.g "confidence $\in [0.9, 1]$".

We can also search for conjunctions of categories, that frequently occur in the list of original association rules (let call them *frequent cedents*).

To find association meta-rules, standard association rule mining algorithms can be used. Encoding of the original rules is thus the key problem in our approach. *Ant* and *Suc* can be encoded either (1) using binary attributes, where each attribute represents one possible literal or (2) using the attributes from the original data set. In both cases we can (or need not) also consider whether the literal occurs in *Ant* or *Suc*. We can thus consider four different representation schemes. So to encode the rule

$$income(high) \wedge balance(high) \Longrightarrow loan(yes)$$

1. when using the encoding based on binary attributes without distinguishing between *Ant* and *Suc*, this rule will be represented using the categories `income_high(true)`, `balance_high(true)` and `loan_yes(true)`.
2. when using the encoding based on original attributes without distinguishing between *Ant* and *Suc*, this rule will be represented using the categories `income(high)`, `balance(high)` and `loan(yes)`.
3. when using the encoding based on binary attributes with distinguishing between *Ant* and *Suc*, this rule will be represented using the categories `Ant_income_high(true)`, `Ant_balance_high(true)` and `Suc_loan_yes(true)`.
4. when using the encoding based on original attributes with distinguishing between *Ant* and *Suc*, this rule will be represented using the categories `Ant_income(high)`, `Ant_balance(high)` and `Suc_loan(yes)`.

Another open question concerning the representation of a rule is whether categories not occurring in the rule should be treated as missing or as negative ones. In the first approach, attributes not used in the rule will be encoded using missing value code. In the second approach, when using the binary representation, categories not used in the rule will get the value false, and when using the original attributes, categories not used in the rule will get a new special value interpreted as not used. Our initial experiments show that using missing value code is more suitable as it will prevent the meta-learning step to generate a great number of meta-rules about non-occurrence of literals in the original rules, this option also corresponds to the original notion of association rules where only items that do occur in the market baskets are taken into consideration.

The selection of a proper representation formalism is closely related to the type of association rules to be analyzed. For apriori like rules, the formalism using the same attributes as for the original data is sufficient. On the contrary, to be able to represent the GUHA like rules (that can contain disjunctions of values of a single attribute or negations of literals) we have to encode each value of each attribute (i.e. each category) as a single binary attribute. This attribute takes value "true" if the encoded category occurs in positive literal, value "false" if the encoded category occurs in negative literal, or value "missing" if the encoded category does not occur in the rule. So the rule

$$balance(high \vee medium) \wedge \neg(unemployed(yes)) \Rightarrow loan(yes) \; / \; sex(male)$$

will be encoded using the categories `balance_high(true)`, `balance_medium (true)`, `unemployed_yes(false)`, `loan_yes(true)` and `sex_male(yes)`.

Quantitative characteristics can be encoded using numerical attributes that must be discretized in advance. There is no difference between the apriori-like and GUHA-like association rules in this encoding.

Anyway, all possible methods of rule encoding will result in building a data table (each rule represented by a single row) that can easily be analyzed using association rule mining algorithm to obtain the meta-rules. The obvious question of this approach is: does such post-processing make sense from the users point of view? We believe that it does, if we answer positively the following questions:

– Do the meta-rules give better insight into the list of "original" association rules?
– Is the list of meta-rules easier to evaluate?

We performed several experiments to find answers to these questions.

3 Experimental Evaluation

To evaluate our ideas, we performed several experiments on data. We carried out the experiments using Weka (a data mining system that is freely available from University of Waikato) [14]. In all of our experiments, we encoded the input rules using the original attributes without distinguishing if the category occurs in *Ant* or in *Suc* (in this case, the representation of the rules has the most similar structure to the original data) and encoding attributes not present in a rule as missing values.

3.1 Running Example

Let us start with a closer look on our running example. When applying the apriori algorithm (the Weka implementation) to the data shown in Table 1, we will obtain (for parameters $minsup = 0.2$, i.e. 2 instances and $minconf = 0.8$) 72 association rules, first 10 of them shown in Table 3. This set of rules has been post-processed in the first series of experiments. We choose the representation of *Ant* and *Suc* based on the original attributes and encode categories not occurring in the rule as "missing". Refer to Table 4 for the encoding of the first ten rules. Notice, that

– the numeric attributes support and confidence have been discretized; we used equifrequent discretization into 2 intervals for both support and confidence in this example,

– we added the attribute true (for technical reasons, to let Weka to find the frequent cedents).

At first we will look for *quantitative* meta-rules. Table 5 shows the listing of all quantitative meta-rules for the parameters $minsup = 0.1$ and $minconf = 0.8$; we intentionally used the same setting of parameters as for the analysis of the original data to compare the number of found rules and meta-rules.

Due to the way how the rules have been encoded for meta-learning, the meta-rules have the same syntax as the original rules. But their meaning is completely different. Recall that the rules are obtained from the original data but the meta-rules are obtained from rules. So the third rule from Table 3 says, that there are 4 clients in the analyzed data with high balance, all of them belonging to category `loan=yes`. But the "same" meta-rule (the rule 10 from Table 5) says that there are 22 rules having the category `balance=high` in *Ant* or in *Suc*, and 18 out of them have also the category `loan=yes` (in *Ant* or in *Suc*). We thus have found a (quite a large) subset of the original rules referring to the same characteristics of the clients.

Table 3. Association rules

```
1. income=high 5 ==> loan=yes 5 conf:(1)
2. loan=no 4 ==> income=low 4 conf:(1)
3. balance=high 4 ==> loan=yes 4 conf:(1)
4. income=high unemployed=no 4 ==> loan=yes 4 conf:(1)
5. income=high sex=female 3 ==> loan=yes 3 conf:(1)
6. income=low sex=female 3 ==> unemployed=yes 3 conf:(1)
7. unemployed=yes loan=no 3 ==> income=low 3 conf:(1)
8. balance=high unemployed=no 2 ==> income=high 2 conf:(1)
9. income=high balance=high 2 ==> unemployed=no 2 conf:(1)
10. income=high balance=high 2 ==> loan=yes 2 conf:(1)
. . .
72. income=high 5 ==> unemployed=no loan=yes 4 conf:(0.8)
```

Table 4. Encoded association rules

Id	true	income	balance	sex	unemp.	loan	abssup	conf
1	t	high	?	?	?	yes	(2.5-inf)	(0.915-inf)
2	t	low	?	?	?	no	(2.5-inf)	(0.915-inf)
3	t	?	high	?	?	yes	(2.5-inf)	(0.915-inf)
4	t	high	?	?	no	yes	(2.5-inf)	(0.915-inf)
5	t	high	?	female	?	yes	(2.5-inf)	(0.915-inf)
6	t	low	?	female	yes	?	(2.5-inf)	(0.915-inf)
7	t	low	?	?	yes	no	(2.5-inf)	(0.915-inf)
8	t	high	high	?	no	?	(-inf-2.5]	(0.915-inf)
9	t	high	high	?	no	?	(-inf-2.5]	(0.915-inf)
10	t	high	high	?	?	yes	(-inf-2.5]	(0.915-inf)

Table 5. Qualitative meta-rules

```
1. income=low loan=yes 7 ==> balance=high 7 conf:(1)
2. unemployed=yes loan=yes 7 ==> balance=high 7 conf:(1)
3. balance=high unemployed=yes 9 ==> income=low 8 conf:(0.89)
4. income=low balance=high 9 ==> unemployed=yes 8 conf:(0.89)
5. balance=high unemployed=no 8 ==> income=high 7 conf:(0.88)
6. income=high balance=high 8 ==> unemployed=no 7 conf:(0.88)
7. balance=medium loan=no 8 ==> unemployed=yes 7 conf:(0.88)
8. balance=medium unemployed=yes 8 ==> loan=no 7 conf:(0.88)
9. loan=no 18 ==> income=low 15 conf:(0.83)
10. balance=high 22 ==> loan=yes 18 conf:(0.82)
11. income=high 26 ==> loan=yes 21 conf:(0.81)
```

Table 6. Quantitative meta-rules

```
1. abssup=(-inf-2.5] 59 ==> conf=(0.915-inf) 59 conf:(1)
2. balance=high 22 ==> conf=(0.915-inf) 22 conf:(1)
3. loan=no 18 ==> conf=(0.915-inf) 18 conf:(1)
4. balance=high loan=yes 18 ==> conf=(0.915-inf) 18 conf:(1)
5. sex=female 15 ==> conf=(0.915-inf) 15 conf:(1)
6. income=low loan=no 15 ==> conf=(0.915-inf) 15 conf:(1)
7. balance=medium 12 ==> abssup=(-inf-2.5] 12 conf:(1)
8. balance=medium 12 ==> conf=(0.915-inf) 12 conf:(1)
9. sex=male 12 ==> abssup=(-inf-2.5] 12 conf:(1)
10. sex=male 12 ==> conf='(0.915-inf)' 12 conf:(1)
11. balance=medium 12 ==> abssup=(-inf-2.5] conf=(0.915-inf) 12 conf:(1)
```

The next step in our running example will be the mining for *quantitative* meta-rules. The input data (encoded rules) remain the same as in the previous step. We again used the parameters $minsup = 0.1$ and $minconf = 0.8$ and we obtained 11 meta-rules shown in Table 6. Like in the set of original rules and the set of qualitative meta-rules, we can again find in the listing a meta-rule dealing with the categories `balance=high` and `loan=yes`. The meta-rule no.4 says, that all original rules having `balance=high` and `loan=yes` in Ant or Suc, have the confidence greater than 0.915.

To be able to use the Weka system also for the last type of analysis, for looking for frequent cedents, we added a dummy category `true=T` to the data that encoded the original association rules. We are thus able to identify frequent cedents from the rules

$$true = T \Longrightarrow Suc,$$

that have sufficiently high confidence. Table 7 shows the 12 respective rules for $minsup = 0.25$ thus showing the cedents Suc that occur in at least 25 percents of the original association rules. We can e.g. see that the category `loan=yes` occurs in more than one half of the original rules.

Table 7. Frequent cedents

```
1. true=t 72 ==> loan=yes 37 conf:(0.51)
2. true=t 72 ==> income=low 30 conf:(0.42)
3. true=t 72 ==> unemployed=no 28 conf:(0.39)
4. true=t 72 ==> unemployed=yes 27 conf:(0.38)
5. true=t 72 ==> income=high 26 conf:(0.36)
6. true=t 72 ==> balance=high 22 conf:(0.31)
7. true=t 72 ==> income=high loan=yes 21 conf:(0.29)
8. true=t 72 ==> income=low unemployed=yes 21 conf:(0.29)
9. true=t 72 ==> income=high unemployed=no 20 conf:(0.28)
10. true=t 72 ==> unemployed=no loan=yes 20 conf:(0.28)
11. true=t 72 ==> loan=no 18 conf:(0.25)
12. true=t 72 ==> balance=high loan=yes 18 conf:(0.25)
```

3.2 Further Experiments

The next set of experiments was carried out on larger (and more realistic) data. We used several data sets from the UCI Machine Learning Repository [13]. The characteristics of the data (number of examples and number of attributes) are summarized in Table 8.

Table 9 summarizes the results of our analysis (both mining association rules and meta-rules) for different data sets. The numbers in the table show the number of found association rules, the number of qualitative meta-rules, the number of quantitative meta-rules, and the number of frequent cedents. To make the numbers comparable, we used the same settings of *minsup* and *minconf* during both learning and meta-learning for corresponding data (*minconf* was in all experiments set to 0.2). The frequent cedents were obtained for $minconf = 0.1$. We used equifrequent discretization into 5 intervals for support and equidistant discretization into 4 intervals for confidence.

The results support our working hypothesis, that the number of meta-rules will be significantly smaller than the number of original rules. Thus the interpretation of meta-rules by domain expert will be significantly less time consuming and difficult compared to the interpretation of the original association rules.

Table 8. Description of used data

Data	no. examples	no. attributes
Brest cancer	286	10
Lenses	24	5
Monk1	123	7
Mushroom	8124	23
Tic-tac-toe	958	10
Tumor	339	18
Vote	435	17

Table 9. Summary of the results

Data	assoc rules	qualitative rules	quantiative rules	frequent cedents
Breast cancer	18742	167	341	80
Lenses	89	13	47	34
Monk1	124	29	30	33
Mushroom	100000	135	109	550
Tic-tac-toe	506	69	30	24
Tumor	100000	234	633	66
Vote	100000	6007	12	150

4 Related Work

The various approaches to post-processing of association rules can be divided into several groups. One group are methods for visualization, filtering or selection of the created rules. This are the standard options in most systems.

Second group contains methods that use some algorithms to further process the rules: clustering, grouping or using some inference methods fits into this group as well as our approach. An application of deduction rules to post-process the results of GUHA method is described in [8]; these rules define allow to remove association rules that are logical consequences of another association rules. Similar idea, but applied to "Agrawal-like" association rules can be found in [12]. This paper also describes clustering of association rules that have the same consequent; the distance between two rules is defined "semantically", i.e. as the number of examples covered only by one of the rules. Both semantical and syntactical (i.e. based on the lists of attribute-value pairs that occur in the rules) clustering of association rules can be found e.g. in [11].

The third possibility is to post-process the rules using some domain knowledge. So e.g. An et all use expert-supplied taxonomy of items for clustering the discovered association rules with respect to the taxonomic similarity ([1]), or Domingues and Rezende ([5]) iteratively scan the itemset rules and updates a taxonomy that is then used to generalize the association mining results.

An additional possibility is to filter out consequences of domain knowledge via application of logic of association rules [8]. This approach is introduced in [10].

5 Conclusions

We present a novel idea of using meta-learning approach to post-process the results of association rule mining. When looking at the two questions from the end of section 2, we can say, that the answer to the first question depends on the domain where association rule mining (and rule post-processing) is applied and that the answer must be given by the domain expert. The answer to the second question can be found in the table 9, where we can see that in all the experiments the number of meta-rules is significantly lower than the number of ordinary rules (and thus should take less time for the domain expert to go through it). Anyway, more experiments and the interpretation of the found meta-rules are necessary to validate the usefulness of the proposed method.

So far we focused on the classical apriori algorithm. Our future work will be oriented on following open issues:

– different types of association rules: the association rules analyzed so far are of the classical form as generated e.g. by the apriori algorithm. Another systems, e.g. LISp-Miner can produce different types of association rules; this brings us to the next issue.
– different types of meta-rules: also the meta-rules created so far are in the form of implications between two conjunctions of attribute-value pairs. When using LISp-Miner for building meta-rules, we can benefit from different types of associations implemented there.
– postprocessing of meta-rules: what will happen if we apply the proposed approach to the meta-rules, i.e. if we perform meta-meta learning?

Acknowledgement

The work is supported by the grant MSMT 1M06014 (from the Ministry of Education of the Czech Republic) and the grant GACR 201/08/0802 (from the Grant Agency of the Czech Republic).

References

1. An, A., Khan, S., Huang, X.: Objective and Subjective Algorithms for Grouping Association Rules. In: Third IEEE Conference on Data Mining (ICDM 2003), pp. 477–480 (2003)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In: SIGMOD Conference, pp. 207–216 (1993)
3. Baesens, B., Viaene, S., Vanthienen, J.: Post-processing of association rules. In: The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2000), Boston, Massachusetts, August 20-23 (2000)
4. Bauer, E., Kohavi, R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning 36(1/2), 105–139 (1999)
5. Domingues, M.A., Rezende, S.O.: Using Taxonomies to Faciliate the Analysis of the Association Rules. In: Second International Workshop on Knowledge Discovery and Ontologies (KDO 2005), ECML/PKDD, Porto (2005)
6. Hájek, P., Havránek, T.: Mechanising Hypothesis Formation - Mathematical Foundations for a General Theory. Springer, Heidelberg (1978)
7. Jorge, A., Poas, J., Azevedo, P.J.: Post-processing Operators for Browsing Large Sets of Association Rules. Discovery Science 2002, 414–421
8. Rauch, J.: Logic of association rules. Applied Intelligence 22, 9–28 (2005)
9. Rauch, J., Šimůnek, M.: An Alternative Approach to Mining Association Rules. In: Lin, T.Y., Ohsuga, S., Liau, C.J., Tsumoto, S. (eds.) Proc. Foundations of Data Mining and Knowledge Discovery. Springer, Heidelberg (2005)

10. Rauch, J.: Considerations on Logical Calculi for Dealing with Knowledge in Data Mining. In: Ras, Z.W., Dardzinska, A. (eds.) Advances in Data Management, pp. 177–202. Springer, Heidelberg (2009)
11. Sigal, S.: Exploring interestingness through clustering. In: Proc. of the IEEE Int. Conf. on Data Mining (ICDM 2002), Maebashi City (2002)
12. Toivonen, H., Klementinen, M., Roikainen, P., Hatonen, K., Mannila, H.: Pruning and grouping discovered association rules. In: Workshop notes of the ECML 1995 Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases, Heraklion, pp. 47–52 (1995)
13. UCI Machine Learning Repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
14. Weka - Data Mining with Open Source Machine Learning Software, `http://www.cs.waikato.ac.nz/ml/weka/`