Using Background Knowledge in GUHA Mining

Martin Ralbovský

Department of Information and Knowledge Engineering,
University of Economics, Prague, W. Churchill Sq. 4, 130 67 Praha 3, Czech Republic
martin.ralbovsky@gmail.com

Abstract. Background knowledge can be used to enhance the data mining process. We discuss the form of background knowledge suitable for data mining with the GUHA method. A new formalization of background knowledge is presented and a tool for background knowledge rules validation is implemented in the Ferda system. We test the new tool to validate rules from medical domain and discuss the usability and possible improvements of the validation method.

Keywords: Background knowledge, GUHA Method, Ferda

1 Introduction

The GUHA method and procedures provide a powerful tool for knowledge discovery in databases. However the comprehension of the method for non data mining experts can be demanding. The experts in the domain need an instrument, which can record their needs and present them to the data mining experts without knowing unnecessary details about mining techniques. This work introduces an innovative look on the topic. *Background knowledge* rules as defined here can help to bridge the gap between the two players by defining data mining tasks. It can also help the data miners to focus on the important relationships in the data and draw conclusions about the data mining techniques. These possibilities are discussed in the work along with experiments to show the usage of the *background knowledge* rules.

The work is structured as follows: Section 2 describes the GUHA method, tools that implement the method and individual procedures used during the work. Section 3 describes the *background knowledge* and suggest its usage as a link between data mining and domain experts. Section 4 introduces new rule formalization with some examples. There are two experiments conducted in Section 5 that use *background knowledge* rules to question the relevance of important quantifiers. Finally, Section 6 puts the work into context of other works dealing with *background knowledge* and Section 7 concludes the work and gives ideas about future work.

2 GUHA Method and Tools

2.1 GUHA Method

The GUHA (General Unary Hypotheses Automaton) is an original Czech method of exploratory data analysis. Since the 60's, the method's theoretical frame (as described e.g. in [4] or [5]) was used over the years as theoretical foundation for several data-mining tools.

GUHA method is implemented by GUHA procedures. GUHA procedure is a program, the input of which consists of the analyzed data and of a simple definition of a large set of relevant patterns. The procedure automatically generates each particular pattern and verifies it against the analyzed data. The output of the procedure consists of all patterns that are true in the analyzed data and are not contained in other patterns.

2.2 GUHA Tools

Apart from the tools presented in this section, several systems implementing GUHA procedures were developed in the past. In recent years, the *LISp-Miner* system[1] has been the most significant GUHA tool. This system has been under development since 1996 at the University of Economics, Prague. The system includes six GUHA procedures in addition to other data preparation and result interpretation modules.

Although the *LISp-Miner* proved to be a successful and stable system suitable for academic data mining, some functionality requirements occurred that the system was not able to fulfill. These requirements concerned among others more coherent (and possibly visual) user environment or bigger modularity.

In 2004, the Ferda project started as an initiative to build a new visual data mining system that would eventually replace the *LISp-Miner* system. Creators (at the Faculty of Mathematics and Physics, Charles University, Prague) succeeded in developing an user friendly visual system with advanced features such as high level modularity, support for distributed computing or reusability of the task setting [6].

The first version of Ferda (version 1.0 and 1.1) was still partly dependent on the old *LISp-Miner* system, because the task boxes (visual elements in Ferda) used the *LISp-Miner* hypotheses generation engine and accessed it through the metabase layer. The new 2.0 version is fully independent on the *LISp-Miner* system and introduces a new hypotheses generating engine with improved task setting abilities (details can be found in [7]).

Because of the advantages of the Ferda system, mainly the high level modularity and the ability to incorporate new independent improvements into existing task setup, we chose Ferda to be the implementation platform for this work[2].

[1] See `http://lispminer.vse.cz`.

[2] In the *LISp-Miner* system, user has to build a standalone application above the existing metabase layer in order to add a module. In Ferda user can add a box module that can take full advantage of partial results of other boxes in the task. More details in [6].

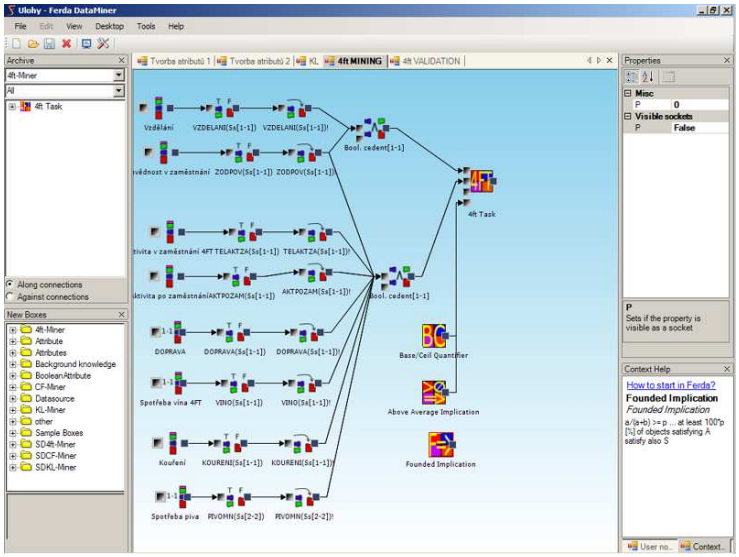

Fig. 1. The Ferda system

2.3 Procedure 4FT

One of the GUHA procedures used in this work is the *4FT* procedure[3]. The procedure extends the historical *ASSOC* procedure for association rules mining and finds the rules in form $\varphi \approx \psi/\chi$ where φ, ψ and χ are *boolean attributes* and $\approx$ is the *4FT quantifier*, as defined in [7] or [13]. φ, ψ and χ are called *antecedent*, *succedent* and *condition* respectively[4]. In this work we use the *founded implication* and *above average* quantifiers, as defined in [11].

2.4 Procedure KL

KL is another GUHA procedure used in this work. The procedure mines for hypotheses of form $R \sim C/\chi$, where R and C are *categorial attributes* representing the row and column attributes, $\sim$ is the *KL quantifier* and χ is the *boolean attribute* representing condition (as defined in [7] or [14]). In this work we used quantifiers derived from the *Kendall quantifier* defined in [9].

[3] Note the terminological difference between the *4FT* procedure and the *4ft-Miner*. The *4ft-Miner* is a module of the *LISp-Miner* system that implements the *4FT* GUHA procedure

[4] In other words, the procedure tests a condition defined by *4FT quantifier* over a four-fold table defined by *antecedent* and *succedent* satisfiing conditing defined by *condition*.

3 Background Knowledge

Background knowledge (or *field knowledge*) is knowledge that comes from the user or a community of users and integrates knowledge, which the community can agree upon and consider it common. The term is vague and different authors define *background knowledge* differently (see Section 6 for details). In the context of GUHA mining, we think of *background knowledge* as a part of domain knowledge, knowledge that is specific to particular domains (medicine, chemistry, etc.). We define *background knowledge* as a set of various verbal rules that are accepted in a specific domain as a common knowledge. The rule can describe functional dependence between quantities, relationship between entities or say something about their behavior.

3.1 *Background knowledge* examples

Examples of such rules can be found in [15], where 46 rules from domains medicine and beer marketing were collected ([15] is the primary source material for this work). Below are presented example rules from the medical domain [5]:

- If education increases, wine consumption increases as well.
- Patients with greater responsibility in work tend to drive to work by car.

Another type of rules displayed below can be found at the beer marketing domain. Unfortunately the source data from this domain are not available. Thus we cannot verify the rules against the source data.

- Younger consumers prefer drought beer.
- Older consumers prefer beer in bottles.
- Cheap brands are better sold in less economically developed regions.
- More expensive brands are better sold during holidays.

3.2 *Background knowledge* usage

Let us distinguish between two types of players present at the data mining process: the domain expert and the data miner. Then *background knowledge* can be used as an effective mean of communication between the two.

The domain expert has extensive knowledge about the domain. He or she understands the meaning of entities or specific levels of measured quantities and can form general rules about behavior or relationships in the domain. Domain experts usually know very little about data mining. On the other hand the data miner does not necessarily need to understand the complexity of the domain. The data miner understands and knows how to use and interpret data mining technique(s).

[5] The rules were taken from the STULONG database. EUROMISE: The STULONG Project `http://euromise.vse.cz/stulong/a-otazky/index.php?page=otazky`. The site is available only in Czech.

The domain expert can form *background knowledge* rules to specify what is interesting or significant for him. The rules are passed to the data miner. Each rule can be transformed into a mining task, which can be run on the examined data afterwards. The results of one or more procedure's run (in the GUHA sense) can confirm or disprove the rule on the examined data. Finding exceptions from a given rule on given data pose also an interesting problem. The results from the data miner can be at last summarized in the form of an analytical report and returned back to the domain expert. All the usage possibilities stated above are possible only with a sound formalization apparatus of *background knowledge* rules.

4 Formalization of Background Knowledge

Background knowledge contains heterogeneous verbal formulations of dependences and relationships in the domain. The relevance and validity of the formulations varies: the relationships in physics are formed exactly by mathematical equations, but for example in sociology they mean only expected behavior or opinion of a group of people. Our aim is to find formalization usable for both domains.

4.1 Qualitative Models

In context of GUHA mining, this work (and the work [15]) is the first attempt to find suitable formalization of *background knowledge* rules. However, in [2] and [10] authors introduced a formalization for induction learning based on qualitative models. We considered this model and found it not to be suitable for GUHA mining, mainly because the strict mathematical requirements of the model. Details can be found in [15] in section 3.2.1.

4.2 Attributes, Validation Literals and Abstract Quantifiers

In order to continue efforts on using *background knowledge* in GUHA mining, we invented a new formalization. The main idea of the new *Formalization with attributes, validation literals and abstract quantifiers* was to make the formalization as close to GUHA terms as possible while still enabling large expressing possibilities of the verbal rule[6]. Because of short format of this article, we present only an overview and an example of the new formalization with a little reasoning. The topic is fully covered in [15], section 3.2.2.

Attribute is the basic term for the new formalization. *Attribute* is defined as a result of domain categorization[7] and is used to create *categorial attributes*, inputs of the *KL* procedure.

[6] As was mentioned in Section 3 the overall theory containing all various types of *background knowledge* does not exist. Therefore we can define a **GUHA background knowledge** as yet another type of *background knowledge*.

[7] There are 4 different means of categorization in the Ferda system, see [6] for details.

Validation literal is a special type of *literal* used for the *background knowledge* validation. *Literal* is a *basic boolean attribute* or its negation. *Basic boolean attribute* is defined by a non-empty subset of the set of categories of an *attribute*. We skip exact definitions, they can be found in [7] or [12]. We define the *literal length* as the size of the categories' subset. *Validation literal* is a literal, which has *literal length* equal to 1.

Abstract quantifier is a generalization of a quantifier or quantifiers of a procedure (*4FT* or *KL*). The idea behind *abstract quantifiers* is to create a "black-box" quantifier: user does not need to fill any numeral parameters of the quantifier. The quantifier is then more suitable for transferring verbal *background knowledge* rules into formalized form.

With all the terms explained, let us see how the formalization is applied to a specific verbal rule `If education increases, wine consumption increases as well.` as presented in Section 3.1. The rule defines relationship between two measurable quantities of a patient. These quantities are stored in the database in the form of columns of a table, so *attributes* can be created. We name the attributes for `education` and `wine consumption` *education* and *wine* respectively.

For this paragraph we will consider only the *KL* procedure. From the Section 2.4 the simplified form of hypotheses for *KL* [8] shortens to $R \sim C$ where R and C are *categorial attributes*, which derive from *attributes*. When `education` and `wine consumption` out of the rule are to be formalized with R and C of the hypothesis, then the part `If ... increases, ... increases as well` could be formalized with a proper *abstract quantifier*. In [15] we called this quantifier *increasing dependence* and implemented it as a special setting of the *Kendall quantifier*, as defined in [9]. With all the knowledge stated above, the rule `If education increases, wine consumption increases as well` can be formally written as $education \uparrow wine$, where $\uparrow$ states for *increasing dependence abstract quantifier*.

We can also define the formalization for the *4FT* procedure. The hypotheses of this procedure consist of *boolean attributes*, therefore *validation literals* need to be used. If we presume correct categorization, out of *attributes education* and *wine* the *validation literals education(HIGH)* and *wine(HIGH)* can be created[9]. Similarly to *KL* formalization we can use *abstract quantifier* to note the dependence. Then the rule `If education increases, wine consumption increases as well` can be formalized as $education(HIGH) \Rightarrow wine(HIGH)$ with a proper *abstract quantifier* $\Rightarrow$[10].

In the beginning of this section, a requirement was given on the formalization to be able to represent various kinds of relationships between the entities of the domain. The *formalization with attributes, validation literals and abstract*

[8] We do not consider the condition.

[9] *Validation literal* allows sign setting. Here both signs are *positive*.

[10] Formalization with the *4FT* procedure does not consider the whole *attributes* but only some categories, thus it is weaker. However there may be situations when it is feasible to use the *4FT* procedure. Details are discussed in [15], Section 3.2.2.

quantifiers clearly fulfills this requirement, because the formalization does not pose any restrictions on the relationships - the relationship is expressed by the *abstract quantifier*.

Background knowledge rule was defined in Section 3 as verbal rule declared by the domain expert. Obviously, not all the rules formed in natural language can be formalized by the formalization we created. Despite this, we think that a substantial part of all the interesting rules can be formalized and the rest could be very hardly analyzed by the implemented GUHA procedures[11] anyway. Below a template for verbal rules that can be formalized is presented:

Under a certain condition defined by a set of *validation literals*, two quantities, entities or states, defined by sets of *validation literals* or *attributes* are in a *relationship*. The relationship is defined by the *abstract quantifier*.

Number of verbal rules that match with the template is high, note also that all the rules presented in Section 3.1 match with the template. This fact itself does not assure success of the formalization, mainly because the real weakness (and strength as well) lies in definition of sound *abstract quantifiers*. We will discuss this matter more in Section 5

5 Experience with Background Knowledge aided GUHA mining

5.1 *Background knowledge* validation tool in Ferda system

A tool for *background knowledge* validation was implemented in the Ferda system for purposes of this work and of work [15]. The tool consist of 10 new box modules that include validation boxes, and boxes enabling the *formalization with attributes, validation literals and abstract quantifiers* construction[12]. The details of the implementation, with proper explanation of the modules and description of validation algorithms can be found in [15].

Although the implementation details of the tool were skipped because of the short format of the article, let us mention the implemented *abstract quantifiers*. For the *KL* procedure, we implemented the *Increasing dependence* and the *Decreasing dependence abstract quantifiers*. As mentioned before, these quantifiers are defined as special settings of the *Kendall quantifier*. For the *4FT* procedure, there exist formal classes of quantifiers and each quantifier belongs to one of that classes[11]. Therefore we chose the *abstract quantifiers* to represent each of the classes. User that has the knowledge of particular quantifiers – does not need the blackbox model (see Section 4.2), can also validate the background knowledge with the normal quantifiers[13].

[11] *4FT* and *KL*

[12] Because the Ferda version 2.0 was still in the development at the time *background knowledge* validation was implemented, we used the version 1.0 We plan to develop the tool also for the version 2.0, with improved possibilities that this version enables.

[13] More details on the topic in [15].

5.2 Validated rules

Out of the 46 gathered *background knowledge* rules, we chose 8 rules for validation from the medical domain[14]. Rules are listed in Table 1.

Number	Rule - right side	left side
1	If education increases	physical activity after work increases as well
2	If education increases	responsibility in work increases as well
3	If education increases	wine consumption increases as well
4	If education increases	smoking decreases
5	If education increases	physical activity in work decreases
6	If education increases	beer consumption decreases
7	Patients with greater responsibility in work	tend to drive to work by car
8	Patients with smaller responsibility in work	tend to use public transport to get to work

Table 1. Validated rules

5.3 Validation with default quantifiers' settings

There are threshold values of parameters defined for each quantifier, which tell us when quantifier's output is significant. We call them *default quantifiers' settings*. These values were set up by an agreement among data mining experts [15]. The first conducted experiment was to verify the values with the aid of the rule validation on the STULONG source data.

For the *KL* procedure *abstract quantifiers*, we chose the *increasing* and *decreasing dependence* with settings of 0.7 and -0.7 value of the *Kendall's* τ_b parameter[16]. For the *4FT* procedure, we wanted to examine settings of the two most used quantifiers from the history of the procedure: the *founded implication* and the *above average* quantifiers. For the *founded implication* quantifier, the default values are 0.95 for the P parameter and 0.05 for the *base* parameter. For the *above average* quantifier, the default values are 1.2 for the P parameter[17] and 0.05 for the *Base* parameter. The *founded implication* and *above average* quantifiers and their default settings are stored in the *Implication* and *Other abstract quantifiers*[18], respectivelly.

[14] We chose the medical domain, because the source data exist. The rules were selected as a sample of the rules that can be mined upon (without changing the database schema).

[15] However, the values are not fixed and can be subject of further discussion

[16] Explanation in [15], Section 5.3.5.

[17] This value corresponds to 0.2 value with the P parameter as defined in [11]. We add 1 to avoid negative computation results.

[18] In the implemented tool of the Ferda system

Rule number	ID	DD	FI	AA
1	YES	x	NO	NO
2	YES	x	NO	NO
3	NO	x	YES	NO
4	x	NO	NO	NO
5	x	NO	NO	YES
6	x	NO	NO	NO
7	x	x	NO	NO
8	x	x	NO	NO

Table 2. Validation of quantifier's settings

Table 2 shows the results of the first experiment. The **ID**, **DD**, **FI** and **AA** stands for *increasing dependence*, *decreasing dependence*, *founded implication* and *above average* quantifiers. **YES** means that the rule was validated with the given quantifier, **NO** means that the rule was not validated and **x** means that the rule was not meaningful for given quantifier.

Before we draw any conclusions from the experiment, let us first state some presumptions about the data source. The data table *Entry*, which was mined upon, contains records about the entry examination of 1417 patients. Because of this number, we consider the data to be statistically significant. We also presume no errors in the data and proper categorization (described in [15]). Finally, when we want to question settings of individual quantifiers, we presume that the *field knowledge* rules are "somehow stored" in the data.

The most interesting result of the experiment the disapproval of all the rules except one with the *founded implication* and also the *above average* quantifier. If a data miner uses the *founded implication* of *above average* quantifiers or derived *abstract quantifiers* with default settings, he probably will not validate any rules - the default settings are too restrictive.

5.4 Finding suitable quantifiers' settings

As the previous section showed, the default settings of a quantifier can be misleading. The next conducted experiment tries to find suitable quantifiers' settings, based on the *background knowledge* rule validation. We gradually decreased the P settings of the *founded implication* and *above average* quantifiers. We did not experiment with the KL quantifiers, because of the complexity of the procedure[19]. With this technique, we could examine more *background knowledge* rules, determine the value of the parameter for each rule and compute the average of the values. New mining with the quantifier can be done with this average value that could discover new relationships in the data.

As we can see in Table 3, the results of the experiment are rather disappointing. For the *founded implication* quantifier, majority of rules had the P value

[19] The results need not to improve merely by changing a parameter of a quantifier. We also need to take the shape of the KL contingency table into consideration.

Rule number	FI	AA
1	0.83	1,03
2	0.72	0.43
3	1	0.68
4	0.32	1.17
5	0.28	1.34
6	0.38	1.17
7	0.16	1.15
8	0.64	1.07

Table 3. Exact quantifiers values

below 0.5. This means that we cannot rely on the quantifier when validating. We got better results for the *above average* quantifier where the P parameter was only twice below 1. However, only once the value exceeded the desired 1.2 value.

To summarize, the experiments with *abstract quantifiers* based on the *founded implication* and the *above average* quantifiers were not successful. These quantifiers have been used as pivot quantifiers for the *4FT* mining. However, as our experiments showed, they failed to validate *background rules* which, according to presumption we made, should be present in the data. Other *abstract quantifiers* should be developed and used in order to get better validation results. Section 7 introduces works that should deal with the topic.

6 Related work

As was mentioned in Section 4.1, there have been other attempts in the past to formalize *background knowledge*, described in [2] and [10]. However, this formalization was no use for GUHA mining.

In [1], authors use *background knowledge* for subgroup discovery. Important part of the work tries to divide *background knowledge* into classes and deals separately with each class. Unfortunately the rules and formalization defined in this work does not belong to any of the classes defined.

In [3], authors developed ideologically similar approach: they used classical association mining in cooperation with a Bayesian network to store the knowledge from domain experts (here called *a priori expert knowledge*) and improved both the association rules mining and the Bayesian network in iterations. This approach is stronger from the methodological point of view (complex methodology is defined) and also enables revision of the domain expert knowledge. However, our *background knowledge* formalization is less restrictive than the Bayesian network and the GUHA procedures offer greater possibilities than the classical association mining.

7 Conclusion and Future Work

The work presents an innovative look on *background knowledge* in the context of GUHA mining. We found techniques to enhance the cooperation between domain expert and the data miner. In order to carry out the techniques, new *formalization with attributes, validation literals and abstract quantifiers* of the *background knowledge* rules was invented. This formalization suits especially the GUHA procedures implemented in the Ferda system.

After implementing a tool for rule validation in the Ferda system, we formalized rules from the medical domain and used them to question the relevance of quantifiers' settings in of the *KL* and *4FT* procedure. Two conducted experiments showed clear inaccuracy in validation of individual rules and a further need to improve the understanding of the quantifiers and their usage.

There are two main directions for future work in the field. The first one covers further investigation of *abstract quantifiers* and conditions under they can be used. New *abstract quantifiers* should be defined, which would have better support of the *field knowledge* rules presented by the domain experts. [8] should be a work on this topic. The second direction is the further improvement of interaction between the data miner and the domain expert. Automated tool should be developed to collect, store and formalize the *background knowledge* rules. In further future, there is a need for automated task generation and result interpretation based on the rules. The *Ever-Miner* system aims to realize these plans[20].

References

1. Atzmueller M., Puppe F.: *A Methodological View on Knowledge-Intensive Subgroup Discovery.* In: S. Staab and V. Svátek (Eds.): EKAW 2006, LNAI 4248, Springer-Verlag 2006, p. 318 – 325
2. Clark P., Matwin S.: *Using Qualitative Models to Guide Inductive Learning.* In: Proceedings 10^{th} International Machine Learning Conference (ML93), pg. 49 - 56
3. Fauré C., Delpart S., Boulicaut J., Mille A.: *Iterative Bayesian Network Implementation by Using Annotated Association Rules.* In: S. Staab and V. Svátek (Eds.): EKAW 2006, LNAI 4248, Springer-Verlag 2006, p. 326 – 333
4. Hájek P., Havel I., Chytil M.: *The GUHA method of automatic hypotheses determination.* Computing 1, 1966, p. 293 – 308
5. Hájek P., Havránek, T.: *Mechanising Hypothesis Formation - Mathematical Foundations for a General Theory.* Springer-Verlag: Berlin - Heidelberg - New York, 1978.
6. Kováč M., Kuchař T., Kuzmin A., Ralbovský M.: *Ferda, New Visual Environment for Data Mining.* Znalosti 2006, Conference on Data Mining, Hradec Králové 2006,p. 118 – 129 (in Czech)
7. Kuchař T.: *Experimental GUHA Procedures*, Master Thesis, Faculty of Mathematics and Physics, Charles University, Prague 2006 (in Czech)
8. Kupka D.: *User support 4ft-Miner procedure for Data Mining.* Master Thesis, Faculty of Mathematics and Physics, Charles University, Prague 2006 (in Czech)

[20] Plans to implement the *Ever-Miner* system are described in [15]

9. Lín V.: *053 Definition of KL quantifiers*. LISp-Miner system documentation (in Czech)
10. Matwin S., Rouge T.: *Explainable Induction with an Imperfect Qualitative Model*. `http://citeseer.ist.psu.edu/matwin95explainable.html`
11. Rauch J.: *Classes of Association Rules - an Overview*. In: Lin T. Y. (ed): Foundations of Semantic Oriented Data and Web Mining. Proceedings of a workshop held in conjunction with 2005 IEEE International Conference on Data Minig
12. Rauch J.: *Logic of Association Rules*. In: Applied Inteligence, Vol. 22, Issue 1, p. 9 – 28
13. Rauch, J., Šimůnek, M.: *An Alternative Approach to Mining Association Rules*. In: Lin T. Y., Ohsuga S., Liau C. J., Tsumomo S. (eds.), Data Mining: Foundations, Methods, and Applications, Springer-Verlag, 2005
14. Rauch, J., Šimůnek, M.: *Mining for Patterns Based on Contingency Tables by KL-Miner - First Experience*. In: ICDM2003 Workshop *Foundations and New Directions of Data Mining*, `http://www.cs.uvm.edu/~xwu/icdm-03.html`
15. Ralbovský M.: *Usage of Domain Knowledge for Applications of GUHA Procedures*, Master Thesis, Faculty of Mathematics and Physics, Charles University, Prague 2006 (in Czech)
16. Svátek V., Rauch J., Ralbovský M.: *Ontology-Enhanced Association Mining*. In: Ackermann, Berendt (eds.). Semantics, Web and Mining, Springer-Verlag, 2006