AN ASSOCIATION RULES SURVEY FOR REDUNDANCY REDUCTION AND DESIRED RULES WITH ONTOLOGY

Dr. V. Vijaya Kumar[1], Dr. P. Suresh Varma[2], G. Srinivas[3]

[1] *Dean, Dept of CSE, GIET, Rajhamundry, E.G.Dist ,A.P*
[2,] *Principal, Dept of CSE, Adikavi Nannayya University, Rajhamundry, E.G.Dist, A.P*
[3] *Research scholor, Pragati Engineering College, E.G.Dist, A.P*

Abstract

In Data Mining generating an association rules is still an important research issue, the usefulness of association rules is strongly limited by the huge amount of delivered rules. To overcome this drawback, several methods were proposed for the reducing the redundant rules and uninteresting patterns. However, being generally based on statistical information, most of these methods do not guarantee that the extracted rules are interesting for the user. Thus, it is crucial to help the decision-maker with an efficient post processing step in order to reduce the number of rules. This paper proposes a new interactive approach to prune and filter discovered rules. The concept of ontology introduced to generalize the data structure and represents data in terms of concepts and then rules schema and applies pruning and filtering techniques can be applied for finding the interesting association rules and to reduce the number of rules to several dozens or less.

Index Terms—*Association rules, Ontology, Pruning and Filtering Techniques*

---***---

1. INTRODUCTION

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. Association rule Mining describes analyzing and presenting strong rules discovered in databases using different measures of interestingness. An association rule is defined as the implication X => Y, described by two interestingness measures support and confidence where X and Y are the sets of items. Furthermore, valuable information is often represented by those rare low support and discovered association rules are unexpected which are surprising to the user. As we increase the support threshold, the more efficient the algorithm**s** are and the more the discovered rules are obvious, and hence, the less they are interesting for the user. As a result, it is necessary to bring the support threshold low enough in order to extract valuable information. Unfortunately, the lower the support is, the larger the volume of rules becomes, making it intractable for a decision-maker to analyze the mining result. Experiments show that rules become almost impossible to use when the number of rules exceeds a limit. Thus, it is crucial to help the decision maker with an efficient technique for reducing the number of rules.

In recent years the sizes of databases has increased rapidly. This has led to a growing interest in the development of tools capable in the automatic extractionof knowledge from data. The term Data Mining, or Knowledge Discovery in Databases, has been adopted for a field of research dealing with the automatic discovery of implicit information or knowledge within databases. The implicit information within databases, and mainly the interesting association relationships among sets of objects, that lead to association rules, may disclose useful patterns for decision support, financial forecast, marketing policies, even medical diagnosis and many other applications. This fact attracted a lot of attention in recent data mining research . Mining association rules may require iterative scanning of large databases, which is costly in processing.Many researchers have focused their work on efficient mining of association rules in databases

2. RELATEDWORK

ASSOCIATION rule mining, introduced in [1], is considered as one of the most important tasks in Knowledge Discovery in Databases [2]. Among sets of items in transaction databases, it aims at discovering implicative tendencies that can be valuable information for the decision-maker. An association rule is defined as the implication X → Y , described by two interestingness measures support and confidence where X and Y are the sets of items and X ∩Y=ϕ.

Apriori

Apriori is the first algorithm proposed in the association rule mining field and many other algorithms were derived from it. Starting from a database, it proposes to extract all association rules satisfying minimum thresholds of support and confidence. It is very well known that mining algorithms can discover a prohibitive amount of association rules; for instance, thousands of rules are extracted from a database of several dozens of attributes and several hundreds of transactions. Furthermore, as suggested by Silbershatz and Tuzilin [3], valuable information is often represented by those rare—low support—and unexpected association rules which are surprising to the user. So, the more we increase the support threshold, the more efficient the algorithms are and the more the discovered rules are obvious, and hence, the less they are interesting for the user.

Drawbacks in the apriori is multiple database scans , it is costly to handle to huge number of candidate sets and number of uninteresting rules

Example : If there are 104 frequent 1-itemsts, the Apriori algorithm will need to generate more than 107 2-itemsets and test their frequencies

Improved Apriori

1. Create a new array PFA[n], the original value for each element is 0; scanning the database, calculating the probability of each itemset A1,A2, …, An respectively and marked by P1, P2, …, Pn .

Let each element of the array PFA[1], PFA[2], …PFA[n] be the P1, P2 ,…, Pn .which refer to the probability of each itemset A1, A2, …, An .

The process for calculating the probability of Ai appearing .

(a) If it is not the end of database ,then read and get the recorder;
(b) If there is an item Ai in the recorder then PFA[i] = PFA[i]+1;
(c) Repeat the above procedure until the end of the database then PFA[i]=PFA[i]/(number of records in the database).

Repeat step (a)(b)(c)to calculate the probability of itemset A1,A2, …, An appearing.

2. Set a minimum value V1 for the probability of Ai appearing, if the probability of A appearing PFA[i] is larger than V1 then itemset Ai is a frequent 1-itemset. so, you get some frequent 1-itemset, let "m" be the number, and PFA[1],PFA[j],…PFA[m] be the probability of 1-itemset appearing respectively.

3 Due to the probability of 1-itemset appearing PFA[1], PFA[j], …PFA[m] ,base on the formula (2),then the probability of any two itemset appeared in one recorder can be evaluated. Set a minimum value V2 for the probability of Ai and Aj appeared synchronously in one record, if the probability is larger than V2 then itemset AiAj is a candidate frequent 2-itemset.otherwise set the value of the probability is zero to predigest the later calculation. Let the element of array PUA2 [i] record the value of candidate frequent 2-itemsets. Let V2 be the minimum probability for candidate frequent 2-itemset, and V3 for candidate frequent 3-itemset, Vk-1 for candidate frequent (k-1)-itemset Set minimum probability Vk-1 :

Vk-1=a*min(PFAk-1[1],PFAk-1[2],…PFAk-1[m])+b*min(PFAk-1[1],PFAk-1[2],…PFAk-1[m])* max(PFAk-1[1],PFAk 1[2],…PFAk-1[m])

4. Recur the above step 1.2.3., from k=2 to n to calculate the probability of k-itemsets A1,A2,… ,Ak appearing in one recorder;

5.Scan the database another time to calculate the support of the candidate frequent itemsets which is the result of step 4.
(a)Create a new array DMA[m] with each element's original value is zero.(m =number of candidate frequent itemsets);
(b)Read and get the recorder of the database until the end of the database.
(c)If there are itemsets Ai,Aj , … ,Ak in any recorder synchronously and Ai≠0,Aj≠0…Ak≠0; then the support for AiAj … Ak DMA[k]= DMA[k]+1. recur the above step(b)(c)to calculate the actual support of every candidate frequent itemsets until the end of the database.

6. Find out the frequent itemsets from the candidate frequent itemsets. If DMA[k] is larger than the minimum support which the user set, then output the frequent itemsets. Step 5.,6. is used to confirm the probability and support of the candidate frequent itemsets which come out by the method of probability evaluation whether satisfy the request of the user.
7. Output the association rule from the result of the step 6.

Fp Growth :

For each frequent item, construct its conditional pattern-base, and then its conditional FP-tree, Repeat the process on each newly created conditional FP-tree, Until the resulting FP-tree is empty, or it contains only one path—single path will generate all the combinations of its sub-paths, each of which is a frequent pattern

FP-growth is faster than Apriori because No candidate generation, no candidate test, Eliminated repeated database scans, Basic operation is counting and FP-tree building (no pattern matching)

Benefits of Fpgrowth algorithm are Preserve complete information for frequent pattern mining, Never break a long pattern of any transaction.Reduce irrelevant info—infrequent items are gone, Items in frequency descending order: the more frequently occurring, the more likely to be shared and Never be larger than the original database

Disadvantage: FP-tree may not fit in main memory, FP-Tree is expensive to build

By applying different pruning techniques, filtering techniques and various postpreprocessing methods are proposed.there are number of Evolutionary algorithms proposed to find the optimal solution and interesting association rules. But still eliminating the redundant rules and finding the intresting rules in still an important research issue. There are number pruning techniques evolved during the years of research.Alogorithms like CLOSET algorithm.

CLOSET:

An Efficient Algorithm for mining Frequent closed Itemsets This approach is an efficient algorithm for mining frequent itemsets with the development of three techniques:

(i) Applying compressed, frequent pattern tree FP-tree structure for mining closed itemsets without candidate Generation.

(ii) Developing a single prefix path compression Technique to identify frequent closed itemsets quickly.

(iii) Exploring a partition based projection mechanism for scalable mining in large databases.

Optimization1: Compress transactional and conditional databases using an FP-tree structure: FPtree compresses databases for frequent itemset mining. An FP tree is a prefix tree structure representing compressed but complete information for a database. Its construction is simple. The transactions with same prefix share the portion of a path from the root. Similarly conditional FP tree can be constructed for conditional databases.

Optimization2: Extract items appearing in every transaction of conditional database: If there exists, a set of items Y appearing in every transaction of the Xconditional database, XUY forms a frequent closed item set if it is not a proper subset of some frequent closed item set with the same support. . This reduces the size of FP-tree because the conditional databases contain less number of items after extraction and also reduces the level of recursion.

Optimization3: Directly extract frequent item sets from FP-tree:

- This allows the program to identify frequent Closed item sets quickly.
- Reduces the size of remaining FP tree to be Examined.
- Reduces the level of recursion.

Optimization4: Prune search branches: Let X and Y are two frequent items with same support. If XCY and Y is closed itemset, there is no need to search for X conditional database because there is no hope to generate frequent item set from there. This reduces the overhead in searching for database.

This paper proposes a new interactive postprocessing approach, to prune and filter discovered rules. First, we propose to use Domain Ontologies in order to strengthen the integration of user knowledge in the postprocessing task. Second, we introduce Rule Schema formalism by extending the specification language proposed by Liu et al. [12] for user beliefs and expectations toward the use of ontology concepts. Furthermore, an interactive and iterative framework is designed to assist the user throughout the analyzing task. The interactivity of our approach relies on a set of rule mining operators defined over the Rule Schemas n order to describe the actions that the user can perform.

3. PROPOSED WORK

Ontology mining for Association rules:

In knowledge engineering and Semantic Web fields, ontologies have interested researchers since their first proposition in the philosophy branch by Aristotle. Ontologies have evolved over the years from controlled vocabularies to thesauri (glossaries), and later, to taxonomies [36]. In the early 1990s, an ontology was defined by Gruber as a formal, explicit specification of a shared conceptualization . By conceptualization, we understand here an abstract model of

ISSN: 2250–3676
[IJESAT] INTERNATIONAL JOURNAL OF ENGINEERING SCIENCE & ADVANCED TECHNOLOGY Volume-2, Issue-5, 1480 – 1486

some phenomenon described by its important concepts. The formal notion denotes the idea that machines should be able to interpret an ontology. Moreover, explicit refers to the transparent definition of ontology elements. Finally, shared outlines that an ontology brings together some knowledge common to a certain group, and not individual knowledge. Several other definitions are proposed in the literature. For instance, in , an ontology is viewed as a logical theory accounting for the intended meaning of aformal vocabulary, and later, in 2001, Maedche and Staab proposed a more artificial-intelligence-oriented definition. Thus, ontologies are described as (meta)data schemas, providing a controlled vocabulary of concepts, each with an explicitly defined and machine processable semantics [16].

Interactive Post mining Process

This framework proposes to the user an interactive process of rule discovery, presented in Fig. 4. Taking into account his/her feedbacks, the user is able to revise his/her expectations in function of intermediate results. Several steps are suggested to the user in the framework as follows:

1. ontology construction—starting from the database, and eventually, from existing ontologies, the user develops an ontology on database items;

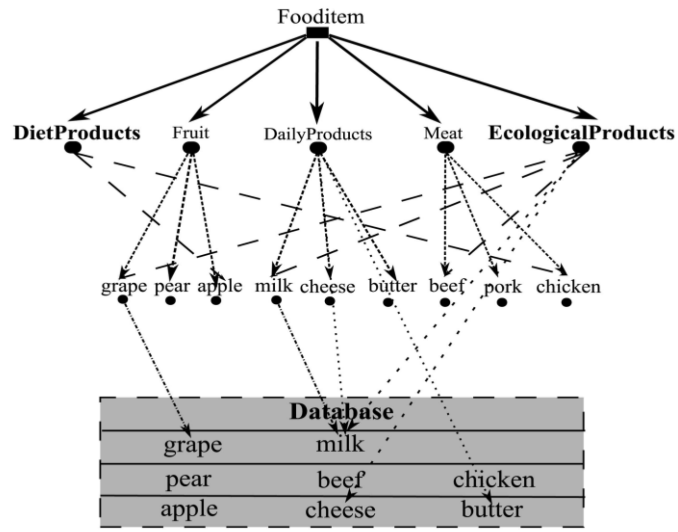
2. defining Rule Schemas (as GIs and RPCs)—the user expresses his/her local goals and expectations concerning the association rules that he/she wants to find;

3. choosing the right operators to be applied over the rule schemas created, and then, applying the operators;

4. visualizing the results—the filtered association rules are proposed to the user;

5. selection/validation—starting from these preliminary results, the user can validate the results or he/she can revise his/her information;
6. we propose to the user two filters already existing in the literature These two filters can be applied over rules whenever the user needs them with the main goal of reducing the number of rules; and

7. The interactive loop permits to the user to revise the information that he/she proposed. Thus, he/she can return to step 2 in order to modify the rule schemas, or he/she can return to step 3 in order to change the operators. Moreover, in the interactive loop, the user could decide to apply one of the two predefined filters discussed in step 6.



Concepts

Domain knowledge, defined as the user information concerning the database, is described in our framework using ontologies. Compared to taxonomies used in the specification language proposed in [12], ontologies offer a more complex knowledge representation model by extending the only is-a relation presented in taxonomy with the set R of relations. In addition, the axioms bring important improvements permitting concept definition starting from existing information in the ontology. In this scenario, it is fundamental to connect ontology concepts C of O {C, R, I,H,A} to the database, each one of them being connected to one/several items of I. To this end, we consider three types of concepts: leaf-concepts, generalized concepts from the subsumption relation (_) in H of O, and restriction concepts proposed only by ontologies.

Rule Schema

The rule schema filter is based on operators applied over rule schemas allowing the user to perform several actions over the discovered rules. We propose two important operators: pruning and filtering operators. The filtering operator is composed of three different operators: conforming, unexpectedness, and exception. We propose to reuse the operators proposed by Liu et al.: conforming and unexpectedness, and we bring two new operators in the postprocessing task: pruning and exceptions.

Operations over Rule Schemas

The rule schema filter is based on operators applied over rule schemas allowing the user to perform several actions over the discovered rules. We propose two important operators: pruning and filtering operators. The filtering operator is composed of three different operators: conforming, unexpectedness, and exception. We propose to reuse the operators proposed by Liu et al.: conforming and unexpectedness, and we bring two new operators in the post processing task: pruning and exceptions. These four operators will be presented in this section. To this end, let us consider an implicative rule schema RS1:(<X → Y>), a nonimplicative rule schema RS2 : (<U; V >), and an association rule AR1 : A → B, where X, Y , U, and V are the ontology concepts, and A and B are the itemsets.

Pruning

The pruning operator allows to the user to remove families of rules that he/she considers uninteresting. In databases, there exist, in most cases, relations between items that we consider obvious or that we already know. Thus, it is not useful to find these relations among the discovered associations. The pruning operator applied over a rule schema, P(RS), eliminates all association rules matching the rule schema. To extract all the rules matching a rule schema, the conforming operator is used.

Conforming.

The conforming operator applied over a rule schema, C(RS), confirms an implication or finds the implication between several concepts. As a result, rules matching all the elements of a nonimplicative rule schema are filtered. For an implicative rule schema, the condition and the conclusion of the association rule should match those of the schema.

Unexpectedness

With a higher interest for the user, the unexpectedness operator U(RS) proposes to filter a set of rules with a surprise effect for the user. This type of rules interests the user more than the conforming one since, generally, a decision-maker searches to discover new knowledge with regard to his/her prior knowledge. Moreover, several types of unexpected rules can be filtered according to the rule schema: rules unexpected regarding the antecedent Up, rules unexpected regarding the consequentUc, and rules unexpected regarding both sidesUb.

For instance, let us consider that the operator Up (RS1) extracts the rule AR1 which is unexpected according to the condition of the rule schema RS1. This is possible if the rule consequent B is conforming to the concept Y , while the condition itemset A is not conforming to the concept X. In a similar way, we define the two other unexpectedness operators.

Exceptions

Finally, the exception operator is defined only over implicative rule schemas (i.e., RS1) and extracts conforming rules with respect to the following new implicative rule schema: X ^ Z →~Y , where Z is a set of items. Conforming to the concept V , then the rule AR1 is conforming with the no implicative rule schema RS2.

Example. Let us consider the implicative rule schema
RS : Fruits → Ecological Products, where
F{Fruits}= f{grape, apple, pear} and
F{Ecological Products} = f{grape; milk};
and I = f{grape, apple, pear, milk, beef} (see Fig. 1 for Supermarket taxonomy). Also, let us consider that the following set of association rules is extracted by traditional techniques:

R1 : grape, beef →milk, pear,
R2 : apple → beef,
R3 : apple, pear, milk → grape,
R4 : grape, pear → apple;
R5 : beef → grape
R6 : milk, beef → grape:

Thus, the operator C(RS) filters the rules R1 and R3,
the operator Up(RS) filters the rules R5 and R6, and the operator U_c(RS) filters the rules R2 and R4. The pruning operator P(RS) prunes the rules selected by the conforming operator C(RS). Let us explain the operator Uc(RS): Uc operator filters the rules whose conclusion item set is not conforming to the conclusion concept of the RS—Ecological Products—and whose condition item set is conforming to the condition concept of the RS—Fruits. The R4 rule is filtered by Uc(RS) because the item set apple does not contain an item corresponding to an Ecological- Products concept, apple 62 f{Ecological, products}, and because the item set grape pear contains at least one item corresponding to a Fruits concept, pear E f{Fruits}.

Filters

In order to reduce the number of rules, three filters integrate the framework: operators applied over rule schemas, minimum improvement constraint filter [24], and itemrelatedness filter [45] Minimum improvement constraint filter [24] (MICF) selects only those rules whose confidence is greater with minimp than the confidence of any of its simplifications.
Example. Let us consider the following three associations rules:
grape, pear → milk (Confidence = 85%)
grape →milk (Confidence = 90%);
pear →milk (Confidence = 83%):

We can note that the last two rules are the simplifications of the first one. The theory of Bayardo et al. tells us that the first rule is interesting only if its confidence improves the confidence of all its simplifications. In our case, the first rule does not improve the confidence of 90 percent of the best of its simplifications (the second rule), so it is not considered as an interesting rule, and it is not selected. The item-relatedness filter (IRF) was proposed by Shekar and Natarajan [45]. Starting from the idea that the discovered rules are generally obvious, they introduced the idea of relatedness between items measuring their semantic distance in item taxonomies. This measure computes the relatedness of all the couples of rule items. We can notice that we can compute the relatedness for the items of the condition or/and the consequent, or between the condition and the consequent of the rule.

CONCLUSION

This paper discusses the problem of selecting interesting association rules throughout huge volumes of discovered rules. The major contributions of our paper are stated below. First, we propose to integrate user knowledge in association rule mining using two different types of formalism: ontologies and rule schemas. On the one hand, domain ontologies improve the integration of user domain knowledge concerning the database field in the postprocessing step. On the other hand, we propose a new formalism, called Rule Schemas, extending the specification language proposed by Liu et al. The latter is especially used to express the user expectations and goals concerning the discovered rules. Second, a set of operators, applicable over the rule schemas, is proposed in order to guide the user throughout the post processing step. Thus, several types of actions, as pruning and filtering, are available to the user. Finally, the interactivity of our framework, relying on the set of rule mining operators, assists the user throughout the analyzing task and permits him/her an easier selection of interesting rules by reiterating the process of filtering rules

REFERENCES:

[1] R. Agrawal, T. Imielinski, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” Proc. ACM SIGMOD, pp. 207-216, 1993.
[2] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.
[3] A. Silberschatz and A. Tuzhilin, “What Makes Patterns Interesting in Knowledge Discovery Systems,” IEEE Trans. Knowledge and Data Eng. vol. 8, no. 6, pp. 970-974, Dec. 1996.
[4] M.J. Zaki and M. Ogihara, “Theoretical Foundations of Association Rules,” Proc. Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD ’98), pp. 1-8, June 1998.
[5] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu, “Mafia: A Maximal Frequent Itemset Algorithm,” IEEE Trans. Knowledge and Data Eng., vol. 17, no. 11, pp. 1490-1504, Nov. 2005.
[6] J. Li, “On Optimal Rule Discovery,” IEEE Trans. Knowledge and Data Eng., vol. 18, no. 4, pp. 460-471, Apr. 2006.
[7] M.J. Zaki, “Generating Non-Redundant Association Rules,” Proc. Int’l Conf. Knowledge Discovery and Data Mining, pp. 34-43, 2000.
[8] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, “Efficient Mining of Association Rules Using Closed Itemset Lattices,” Information Systems, vol. 24, pp. 25-46, 1999.
[9] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hatonen, and H. Mannila, “Pruning and Grouping of Discovered Association Rules,” Proc. ECML-95 Workshop Statistics, Machine Learning, and Knowledge Discovery in Databases, pp. 47-52, 1995.
[10] B. Baesens, S. Viaene, and J. Vanthienen, “Post-Processing of Association Rules,” Proc. Workshop Post-Processing in Machine Learning and Data Mining: Interpretation, Visualization, Integration, and Related Topics with Sixth ACM SIGKDD, pp. 20-23, 2000.
[11] J. Blanchard, F. Guillet, and H. Briand, “A User-Driven and Quality-Oriented Visualization for Mining Association Rules,” Proc. Third IEEE Int’l Conf. Data Mining, pp. 493-496, 2003.
[12] B. Liu, W. Hsu, K. Wang, and S. Chen, “Visually Aided Exploration of Interesting Association Rules,” Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD), pp. 380-389, 1999.
[13] G. Birkhoff, Lattice Theory, vol. 25. Am. Math. Soc., 1967.
[14] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, “Discovering Frequent Closed Itemsets for Association Rules,” Proc. Seventh Int’l Conf. Database Theory (ICDT ’99), pp. 398-416, 1999.
[15] M. Zaki, “Mining Non-Redundant Association Rules,” Data Mining and Knowledge Discovery, vol. 9, pp. 223-248, 2004.
[16] A. Maedche and S. Staab, “Ontology Learning for the Semantic Web,” IEEE Intelligent Systems, vol. 16, no. 2, pp. 72-79, Mar. 2001.
[17] B. Liu, W. Hsu, L.-F. Mun, and H.-Y. Lee, “Finding Interesting Patterns Using User Expectations,” IEEE Trans. Knowledge and Data Eng., vol. 11, no. 6, pp. 817-832, Nov. 1999.
[18] I. Horrocks and P.F. Patel-Schneider, “Reducing owl Entailment to Description Logic Satisfiability,” J. Web Semantics, pp. 17-29, vol. 2870, 2003.

[19] J. Pei, J. Han, and R. Mao, “Closet: An Efficient Algorithm for Mining Frequent Closed Itemsets,” Proc. ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery, pp. 21-30, 2000.
[20] M.J. Zaki and C.J. Hsiao, “Charm: An Efficient Algorithm for Closed Itemset Mining,” Proc. Second SIAM Int’l Conf. Data Mining, pp. 34-43, 2002.
[21] M.Z. Ashrafi, D. Taniar, and K. Smith, “Redundant Association Rules Reduction Techniques,” AI 2005: Advances in Artificial Intelligence – Proc 18th Australian Joint Conf. Artificial Intelligence pp. 254-263, 2005.
[22] A. Bellandi, B. Furletti, V. Grossi, and A. Romei, “Ontology- Driven Association Rule Extraction: A Case Study,” Proc. Workshop Context and Ontologies: Representation and Reasoning, pp. 1-10, 2007.
[23] R. Natarajan and B. Shekar, “A Relatedness-Based Data-Driven Approach to Determination of Interestingness of Association Rules,” Proc. 2005 ACM Symp. Applied Computing (SAC), pp. 551- 552, 2005.
[24] A.C.B. Garcia and A.S. Vivacqua, “Does Ontology Help Make Sense of a Complex World or Does It Create a Biased Interpretation?” Proc. Sensemaking Workshop in CHI ’08 Conf. Human Factors in Computing Systems, 2008.
[25] A.C.B. Garcia, I. Ferraz, and A.S. Vivacqua, “From Data to Knowledge Mining,” Artificial Intelligence for Eng. Design, Analysis and Manufacturing, vol. 23, pp. 427-441, 2009.
[26] L.M. Garshol, “Metadata? Thesauri? Taxonomies? Topic Maps Making Sense of It All,” J. Information Science, vol. 30, no. 4, pp. 378-391, 2004.
[27] I. Horrocks and P.F. Patel-Schneider, “A Proposal for an owl Rules Language,” Proc. 13th Int’l Conf. World Wide Web, pp. 723-731, 2004.
[28] W.E. Grosso, H. Eriksson, R.W. Fergerson, J.H. Gennari, S.W. Tu, and M.A. Musen, “Knowledge Modeling at the Millennium (the Design and Evolution of Protege-2000),” Proc. 12th Workshop Knowledge Acquisition, Modeling and Management (KAW ’99), 1999.
[29] M.-A. Storey, N.F. Noy, M. Musen, C. Best, R. Fergerson, and N. Ernst, “Jambalaya: An Interactive Environment for Exploring Ontologies,” Proc. Seventh Int’l Conf. Intelligent User Interfaces (IUI ’02), pp. 239-239, 2002.