SEWEBAR-CMS: A System for Postprocessing Association Rule Models

Tomáš Kliegr, David Chudán, Andrej Hazucha, Jan Rauch

Faculty of Informatics and Statistics, VŠE Praha,
Nám. Winstona Churchilla 4, 130 67 Praha 3, Czech Republic,
tomas.kliegr@vse.cz, david.chudan@vse.cz, andrej.hazucha@vse.cz,
rauch@vse.cz

Abstract. The principal problem of the association rule (AR) mining task is the selection of rules that might be interesting for the domain expert from the many rules typically generated by the software. SEWEBAR-CMS is a Joomla!-based Content Management System for post-processing AR models that supports the data analyst in this effort. The input for the system are AR models in GUHA-extended PMML AR model and machine-readable background knowledge elicited from domain experts within the CMS. PMML and background knowledge are converted to auto-generated reports with XSLT2HTML transformation and presented as CMS documents. They are also semantized according to the Association Rule Mining Ontology, interlinked, and stored in an external Ontopia knowledge base, which uses the Topic Map semantic web formalism. Queries issued from the CMS against Ontopia in the tolog language are used to select discovered ARs that are in some interesting relationship (e.g. exception, confirmation) with the background knowledge. The data analyst presents the mining results to the domain expert through the analytical report that blends in query results with free text and fragments of the auto-generated reports.

1 Introduction

Analytical report is a free-text document describing various elements of an association rule mining task: the data, pre-processing steps, task setting and particularly the *potentially interesting* discovered association rules. The presented system addresses the well-known problem of the association rule mining task – selection of the *potentially interesting rules*. Typically, too many rules are produced and it is difficult for the data analyst to select which rules might be interesting for the domain expert.

SEWEBAR-CMS system[1] is a web-based content management system for presentation of association rule mining results that addresses this problem by letting the domain expert input his knowledge of the problem using an intuitive user interface, which requires little interaction. The elicited information is

[1] Semantic Web and Analytical Reports - Content Management System

translated into background association rules that are subsequently matched with discovered association rules.

This paper is organized as follows. In Section 2, we give a brief overview of the systems architecture, typical workflow and the exchange formats used, Section 3 briefly describes the background knowledge elicitation formalism and user interfaces. Section 4 presents integration of SEWEBAR-CMS with external semantic knowledge base, which performs matching of mined rules with background knowledge. Section 5 demonstrates the framework on an example of post-processing the results of mining from a synthetic financial dataset. The conclusions discuss the interoperability with mainstream data mining systems.

It should be noted at this point that it is out of the scope of this paper to describe the association rule mining task, a concise definition can be found in most data mining textbooks.

2 Architecture, Workflow and Formats

SEWERAR-CMS serves as a communication hub between the data analyst, domain expert, the association rule mining software and post-processing systems such as the semantic knowledge base. The data mining system (Figure 1A) and the post-processing system (Figure 1D) run on an arbitrary platform and are connected with SEWEBAR-CMS via a SOAP/XML-RPC/REST web service. SEWEBAR-CMS is built on top of the PHP-based open source Joomla![2] CMS, one of the most popular open source CMS systems at the time of writing. Joomla!'s advantages include object-oriented architecture, thousands of available extensions and an active community.

2.1 Workflow

While the system does not enforce a specific workflow per se, we describe here the recommended one. Figure 1B shows the domain expert inputing her background knowledge about the dataset and patterns (aka association rules) in the mined domain including patterns that have been refuted and conjectures that she would like the data analyst to explore. The elicited knowledge is saved into the Background Knowledge Exchange Format (BKEF).

Referring to the background knowledge, which is presented in the CMS as an XSLT-generated HTML document, the data analyst starts mining the data. First, he uses the background knowledge to pre-process the data and then for setting up the mining task. Figure 1A shows the mining result exported into PMML (GUHA AR PMML) format from the data mining system and sent via a web-service to SEWEBAR-CMS. In Figure 1C the PMML document is XSLT-transformed to an auto-generated report . The data analyst usually performs and exports multiple experiments varying preprocessing steps and settings. Within the CMS data analyst then initiates another export, which sends the GUHA AR

[2] `http://www.joomla.org`

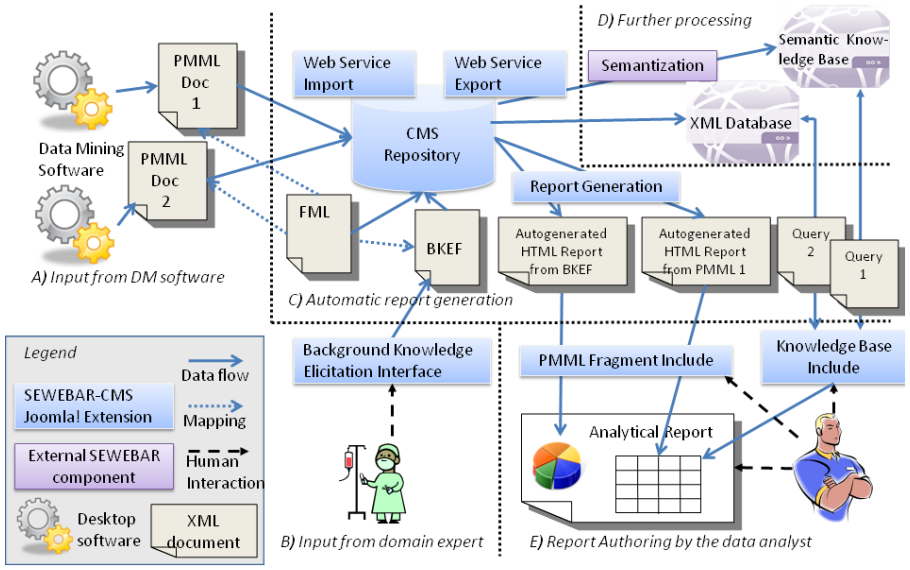

Fig. 1. Framework outline

PMML documents and BKEF documents to an external knowledge base. In this paper, we focus on the case when this is a semantic knowledge base, but it can be e.g. an XML database.

Eventually in Figure 1E, the data analyst creates a textual report conveying the results of mining to the domain expert. In the report, the analyst can mix fragments from the autogenerated document with background knowledge and autogenerated PMML reports. These fragments carry machine-readable information, so that they can be updated when the underlying PMML or BKEF changes. Most importantly, the domain expert issues queries against the knowledge base for rules that are in some interesting relationship with background knowledge patterns. Matching rules are included into the analytical report in the same updatable fashion as report fragments.

A Field Mapping Language (FML) is a simple specification intended to map BKEF and PMML documents (Fig. 1C). The tools for generation of FML documents are under active development. In the current system, we use manually created FML documents. In the following two subsections, we will describe in greater detail the GUHA AR PMML and BKEF formats, which are of central importance for the system.

2.2 GUHA AR PMML and BKEF

GUHA AR PMML [6] is an extension of the PMML (Predictive Markup Modeling Language) 4.0 Association Rule model, which is the industry standard for representing the output of association rule mining systems. The purpose of the

GUHA extension is to increase the expressivity of the model so that it can be used with a broader range of Association Rule mining software. A GUHA AR PMML document contains description of the input data set, the transformations performed on its fields, association rule mining settings and most importantly the set of discovered association rules. The most marked differences between classical and GUHA ARs are described in [6], however, these differences are not important for this demo.

BKEF (Background Knowledge Exchange Format) documents [5] describe the knowledge of the *domain expert* relating to the data being analyzed. This specification was proposed within the SEWEBAR-CMS to satisfy the need for a formal description of background knowledge that would be interoperable with PMML. A BKEF document consists of two distinct pieces of information: the mandatory definition of *Meta-attributes* and the optional *Patterns*. The *Meta-attribute* [7] is an abstraction of a data field in the input dataset, which the domain expert uses to convey information about data fields such as suggestions for value transformations. Metaattributes can then be used to construct patterns, such as background association rules. Background association rules have a similar structure as association rules, but are expressed by the domain expert.

3 Background Knowledge Elicitation

SEWEBAR-CMS contains two Joomla! extensions for elicitation of background knowledge. The Metaattribute Editor is intended for elicitation of user background knowledge on individual data fields, this includes particularly outliers, discretization hints and significant values. Influence Editor gathers background knowledge association rules through mutual influences, an easy to understand graphical representation. One input mutual influence can be transformed to multiple association rules.

For example, if the domain expert wants to convey that with increasing values of *Age*, there is a smaller chance of a *good* value of *Quality*, he will use the mutual influence of the type $Age \uparrow^{-} Quality$. Such a pattern can correspond to multiple (background) association rules, for example

```
BAR1: Age(<65;75))=>Quality(Bad)  BAR2: Age(<55;65))=>Quality(Bad)
```

The transformation of mutual influences to background association rules can be specified either manually using background association rules, or automatically as described in [7]. Note that for the mutual influence of type $A \uparrow^{-} B$, [7] assumes that the B attribute is binary. For the attribute Quality={Good, Bad}, this can be met by putting quality(Good) $\leftrightarrow$ quality_good(true) and quality(Bad) $\leftrightarrow$ quality_good(false).

4 Semantic Knowledge Base Integration

BKEF, PMML and FML files stored in the CMS database are transformed into instances of concepts of the Background Knowledge Ontology [5], FML to

Schema Mapping Ontology and GUHA AR PMML into the Association Rule Ontology [3] and stored in the OKS. Essentially, during the semantization elements in the input XML document are transformed into instances of similarly named concepts, attributes are converted into occurrences, though it should be noted that this is not one-to-one mechanistic transformation.

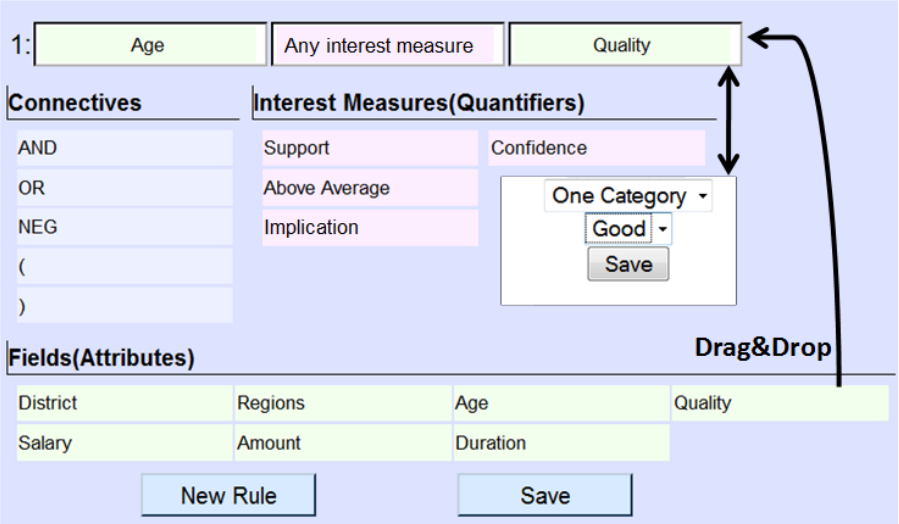

Fig. 2. Association Rule Query Designer screenshot

The fact that both discovered and background knowledge is semantized and interlinked can be utilized in tolog queries. tolog (written with lower case) is a Topic Map query language that combines SQL with Prolog. The system allows either entering a tolog query directly, or using the visual *Association Rule Query Designer* (ARQD) [2] depicted on Fig. 2. The SEWEBAR-CMS includes the Knowledge Base Include extension [2], which initializes ARQD with data field names from a specific knowledge base, lets the user define the query in ARQD, executes the query against the knowledge base and finally includes the query result into the analytical report.

5 Case Study: Financial Dataset

The case study demonstrates the SEWEBAR-CMS system on an updated version of the Financial Data Set first introduced in the PKDD'99 Discovery Challenge (`http://lisp.vse.cz/pkdd99/`). The Financial Dataset consists of 8 tables describing the operations of bank customers. Among the 6181 customers the aim is to identify subgroups with high occurrence of bad loans.

For the mining schema we used the columns `salary`, `status`, `district`, `amount`, `duration` and `quality`; all columns come from the `Loans` table.

5.1 Background Knowledge

The hypothetical financial expert uses the elicitation interface presented in Section 3 to input background knowledge relating to the dataset. For example, she uses the Metaattribute Editor to enter two preprocessing hints for the *district* field, which contains the location of the client's permanent residence. The *Region* preprocessing hint recommends to decrease the granularity of the field by grouping districts (default value from the database) into regions. In this way, the number of distinct values in the field drops from 77 to 14. The *District* hint recommends to leave the values on the district granularity.

After entering information on individual data fields, the expert proceeds to the Influence Editor and uses it to enter confirmed mutual influences. By selecting a proper arrow symbol in the attribute matrix, the following pieces of backround knowledge are entered:

- **BK1** With increasing age, the relative frequency of good loans decreases.
- **BK2** With increasing salary, the relative frequency of bad loans decreases.

5.2 Data Mining

The data analyst loads the data matrix into the LISp-Miner data mining system[3] (`http://lispminer.vse.cz`). First, the data need to be preprocessed. The data-miner performs automatic binning for data fields, where there is no corresponding metaattribute in the background knowledge, or it does not specify a preprocessing hint. The latter is the case of the *Age* attribute, where values are equidistantly binned into intervals $\langle 25; 35)$, $\langle 35; 45)$ ... Remaining fields such as district are preprocessed based on the background knowledge, out of the two possible preprocessing hints for the district field, the *District* hint is chosen.

After the preprocessing is finished, the data analyst formulates the following association rule mining task: Age(interval 1 - 2) $\wedge$ $Regions$(subset 1 - 1) $\wedge$ $Duration$(subset 1 - 1) $\wedge$ $Amount$(subset 1 - 1) $\wedge$ $Salary$ (cut 1 - 2) $\Rightarrow$ $Quality$ (subset 1 - 1). The interest measure threshold was set as follows $minConf = 0.9$ and $base = 100$. Explanation of the mining setting is out of the scope of this paper, details can be found e.g. in [8].

The data mining system finds 258 rules in the data, storing the result as Task 1 PMML model. Although this result might contain valuable knowledge, the rules are too numerous. The analyst therefore decides to reiterate the mining with altered setting hoping to obtain rules that concern only the bad quality loans and to cut down the number of rules by applying the *Region* preprocessing hint and by removing columns *amount* and *duration*. In this revised Task 2 PMML model only 7 rules were found.

5.3 Authoring the report

The resulting mining model is exported to GUHA AR PMML and sent via a web service to the SEWEBAR-CMS. The model is there immediately visualized

[3] SEWEBAR-CMS can be viewed as a web-based spin off from the LISp-Miner project

using an XSLT transformation into a human readable report. Similarly, the knowledge obtained from the domain expert is stored as a machine-readable BKEF document and visualized with an XSLT transformation into an HTML page of the CMS system. These autogenerated reports are intended for reference by the data analyst as they are necessarily too verbose for the end user. The two PMML documents as well as the BKEF document, which was created based on the domain expert's input, are semantized and merged into one topic map.

The end user is presented an *analytical report*, which is a concise document created by the data analyst that sums all the information about the data mining task and highlights the most interesting discovered association rules.

To make the report concise, the analyst needs to highlight in it the discovered rules, which he deems interesting with respect to the background knowledge. Optionally, the report may also contain a section including rules that are clearly confirming the expert's knowledge. Since the manual evaluation of all the 265 rules discovered in both tasks would be too laborious, the expert uses the Semantic Knowledge Base to perform the search.

In the first query, the data miner tries to find rules that *comply* with the first piece of the background knowledge conveyed by the domain expert (see Section 5.1): *With increasing age, the relative frequency of good loans decreases.* The data analyst therefore decides to search for rules that involve young age and good loan quality. He uses the Association Rule Query Designer (ARQD) to formulate a query, which will find all rules containing age group $\langle 25; 35)$ in the antecedent and good loan quality in the consequent. Figure 2 presents the screenshot of the ARQD interface with the query formulated. The query finds 40 such rules, several discovered rules are listed:

```
1 Age(<25;35)) & Amount(<0;100000)) & Duration(12)=>Quality(Good)
20 Age(<25;45)) & Duration(12) & Salary(medium)=>Quality(Good)
29 Age(<25;35)) & Amount(<0;100000)) & Salary(low, medium)=>Quality(Good)
40 Age(<25;35)) & Amount(<0;100000)) & Salary(medium, high)=>Quality(Good)
```

The analyst puts this query result into the part of the report, which deals with confirming the domain expert's knowledge.

With the second query, the analyst wants to find discovered rules that might be surprising for the expert. According to the two pieces of background knowledge combined, old people with very low salaries should have bad loan quality. The analyst therefore formulates a query searching for rules that capture the opposite relationship: old people (age in "$\langle 55; 65)$") with very low salary and *good* loan quality.

There are 27 rules returned and inserted into the report. The analyst inspects the result and discards the obvious rules, particularly the ones that limit loan amount to the lowest bands. After doing this, the report contains the following two rules

```
1 Age(<55;65)) & Duration(24) & Salary(very low, low)=>Quality(Good)
2 Age(<55;65)) & Duration(12) & Salary(very low, low)=>Quality(Good)
```

Into the same section, the analyst includes a fragment of the BKEF document containing the contradicting pieces of background knowledge.

Note that the queries could be created automatically from background knowledge if an FML mapping between BKEF and PMML models exists. This approach is in detail described in [3].

6 Summary

SEWEBAR-CMS introduces, to the best of our knowledge, the first systematic solution to post-processing association rule mining results that exploits semantic web technologies. The framework is built upon proven standards and technologies such as XML and content management systems. The proposed data mining ontology is designed with respect to the industry standard PMML specification, which should foster adoption of the framework among data mining practitioners.

Since the GUHA AR PMML, the framework input format, is based on PMML, the prototype SEWEBAR-CMS implementation can be easily adapted to consume results from other DM tools that use the industry standard apriori algorithm [6]. The ontology, Joomla! extensions and XML schemas are available online at `http://sewebar.vse.cz/`. The website also presents a SEWEBAR-CMS demo including short video showcases.

Acknowledgment The SEWEBAR-CMS system is a result of contribution of a number of undergraduate and graduate students and researchers. Particularly, we would like to recognize the contribution of the following colleagues: Jakub Balhar, Vojtěch Jirkovský, and Stanislav Vojíř. This work was supported by grant IGA 15/2010 of the University of Economics, Prague and by Institutional Funds of the Faculty of Informatics and Statistics of the University of Economics, Prague.

References

1. DMG: PMML 4 Specification, Online: `http://www.dmg.org/pmml-v4-0.html`
2. Hazucha A., Balhar, J., Kliegr, T.: A PHP library for Ontopia-CMS Integration . TMRA 2010. University of Leipzig, 2010. To Appear
3. Kliegr, T., Ovečka M., Zemánek, J.: Topic Maps for Association Rule Mining. In: TMRA 2009, University of Leipzig, November 11-13, 2009.
4. Kliegr M., Ralbovský M., Svátek, V, Šimůnek M., Jirkovský V., Nemrava J., Zemánek J.: Semantic Analytical Reports: A Framework for Post-Processing Data Mining Results. In: ISMIS'09. Springer Verlag, LNCS, 2009, 88-98.
5. Kliegr T., Svátek, V, Šimůnek M., Stastný D., Hazucha A.: An XML Schema and a Topic Map Ontology for Formalization of Background Knowledge in Data Mining. In: IRMLeS-2010, Heraklion, Crete, Greece, May 2010. Online `http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-611/`
6. Kliegr T., Rauch, J.: An XML Format for Association Rule Models Based on GUHA Method. In: RuleML-2010, 4th International Web Rule Symposium, Washington, DC, USA, October 2010, Springer LNCS. To appear.
7. Rauch J.: Considerations on Logical Calculi for Dealing with Knowledge in Data Mining. In: Advances in Data Management. Studies in Computational Intelligence, Volume 223/2009, Springer 2009.
8. Rauch, J., Šimůnek, M.: An Alternative Approach to Mining Association Rules. In: Lin T Y, Ohsuga S, Liau C J, and Tsumoto S (eds): Foundations of Data Mining and Knowledge Discovery. Springer-Verlag, 2005, 219-239.