Business Rule Learning with Interactive Selection of Association Rules

Stanislav Vojíř, Přemysl Václav Duben, and Tomáš Kliegr

Department of Information and Knowledge Engineering
Faculty of Informatics and Statistics
University of Economics, Prague, Czech Republic,
{stanislav.vojir|xdubp00|tomas.kliegr}@vse.cz

Abstract. This paper presents the implementation of a classification system based on learning of association rules in conjunction with Drools rule engine. The rules are interactively discovered with a web-based data mining system EasyMiner.eu. The rules are approved and edited by the domain expert before they are deployed for classification.

Keywords: association rules, business rules, drools, classification

1 Introduction

Today, there is an increasing demand for decision support systems (DSS). The penetration of DSS solutions to many domains is stifled by the fact that building a DSS requires a significant amount of time from users, who need to be not only domain experts, but also skilled knowledge engineers. While algorithmic support for decision making support is a complex endeavour, many problems can be cast as a classification task for which a model can be induced automatically from data. Example of such tasks include answering questions such as: ”Is this product appropriate for this customer?”, or ”Is the user interested in this media content?”

While a plethora of machine learning algorithms exist, most of them produce “black box” models which are difficult to comprehend and manipulate. Decision tree induction and rule induction suffer from this problem, perhaps, the least. The advantage of these models that the they consists of individual units of knowledge (rules),[1] which can be separately viewed, evaluated and manipulated – deleted or even edited.

This paper presents a complete implementation of a prototype decision support system, which:

- automatically learns decision rules from data,
- allows the domain expert to edit the resulting rule base,
- deploys the rules to a Business Rule Management System (BRMS), which can apply the rules to score incoming objects.

[1] Decision trees can be converted to rules.

The system is built on top of the data mining system EasyMiner and a BRMS system Drools Expert.

This paper is organized as follows: Section 2 presents the motivation and related work. Section 3 describes individual steps of the preparation process of a classification rule set. Section 4 covers the implementation. The conclusion presents the technical details of the demo and with an outlook for further work.

2 Motivation and Related work

We motivate our technological solution with the following use case.

Use case example. Alice works as a business analyst for a small community bank. She is assigned a task to prepare a credit scoring application. She has at her disposal a dataset describing loans awarded and the rating of the loan after a one year period. Applying a rule learning algorithm yields a classification model consisting of 100 rules.
Looking at the rules, Alice is not absolutely satisfied. She deletes some of the rules, which seem to be artifacts of the data, and introduces new rules that represent well-known relations between client demographics and rating that were omitted by the learning algorithm. After verifying that the accuracy of the edited rule set increased on a holdout dataset, she deploys the model.

If Alice would have to rely on commodity rule learning systems, she would have to probably export the rule learning results into the PMML format[2], edit the resulting XML, and import the result to a scoring engine. The system presented in this paper allows to accomplish the task through a chain of web-based graphical tools, which are built on top of industry standard BRMS software and open formats.

Research of "business rules" generation from data mining results is a current, yet not very exposed, research topic. To the best of our knowledge, the only publicly available software solution to business rule learning is *Rule learner* [5]. Rule learner is a part of *OpenRules Decision Management System*[3], which uses *Weka* system[4] to perform rule learning. The learned rules are presented in the form of decision tables in Microsoft Excel worksheets.

However, there are several experimental systems developed in the machine learning community with related functionality. *MIME Framework* [1] is a desktop application built also on top of Weka, which provides interactive rule mining and filtering of discovered rules.

The EasyMiner system [8], used in our solution, combines user-defined constraints on the search space with interactive mining. The interactive nature of MIME and EasyMiner systems can help to address the excessive number of

[2] `http://www.dmg.org/pmml-v4-0-1.html`
[3] `http://www.openrules.com`
[4] `http://www.cs.waikato.ac.nz/ml/weka/`

rules that are typically on the output of rule learning. Additionally, in [3] we have shown that the CBA database pruning algorithm can reduce the number of rules output by EasyMiner significantly, without adversely impacting the quality of the classifier.

BigML[5] is perhaps the most well-known representative of web-based tools. BigML uses decision tree induction, but allows to export the resulting tree also into a rule set.

Here we present EasyMiner-BR, as an extension of the EasyMiner system. Compared to other software packages described above, EasyMiner-BR covers the entire workflow in a single environment. Within one web application, the user can launch multiple rule learning tasks, export selected rules to the knowledge base (business rule base), edit the saved rules and apply the rules on test data.

The brCBA rule learning algorithm used in EasyMiner-BR for building classifiers from association rules is evaluated and described in detail in [3]. This paper presents the implemented software solution in a demo form. The contribution, with respect to the published results [7,8], is in the integration of the EasyMiner system with the brCBA algorithm, and with the editor of classification "business" rules.

3 Rule base preparation workflow

The process of preparing a rule base in EasyMiner-BR can be divided into several consecutive steps which are presented on Figure 1: 1) data preparation, 2) association rule learning, 3) rule selection, 4) classification model testing, 5) rule set editing and 6) deployment of rules to BRMS. These steps, which can be performed with a standard web browser, are described in the following.

3.1 Data preparation

The user starts with a dataset for training of the classification model. Typically, the dataset is divided into a training and a testing part. Currently, the system does not support an automatic split of the input table into a testing and training subset, the user needs to perform this task with other tools.

The dataset for model training (association rule learning) can be stored in a CSV file or in a MySQL database. The user inputs connection information or uploads a CSV file. While uploading, the user is able to select file encoding, separators etc.

For association rule learning, it is usually necessary to discretize numerical columns. The system supports several types of preprocessing. For numerical columns, the user can define preprocessing using intervals. For nominal as well as numerical columns, the user can merge values by enumeration, creating *bins of values*. More details can be found in [8].

[5] `http://bigml.com`

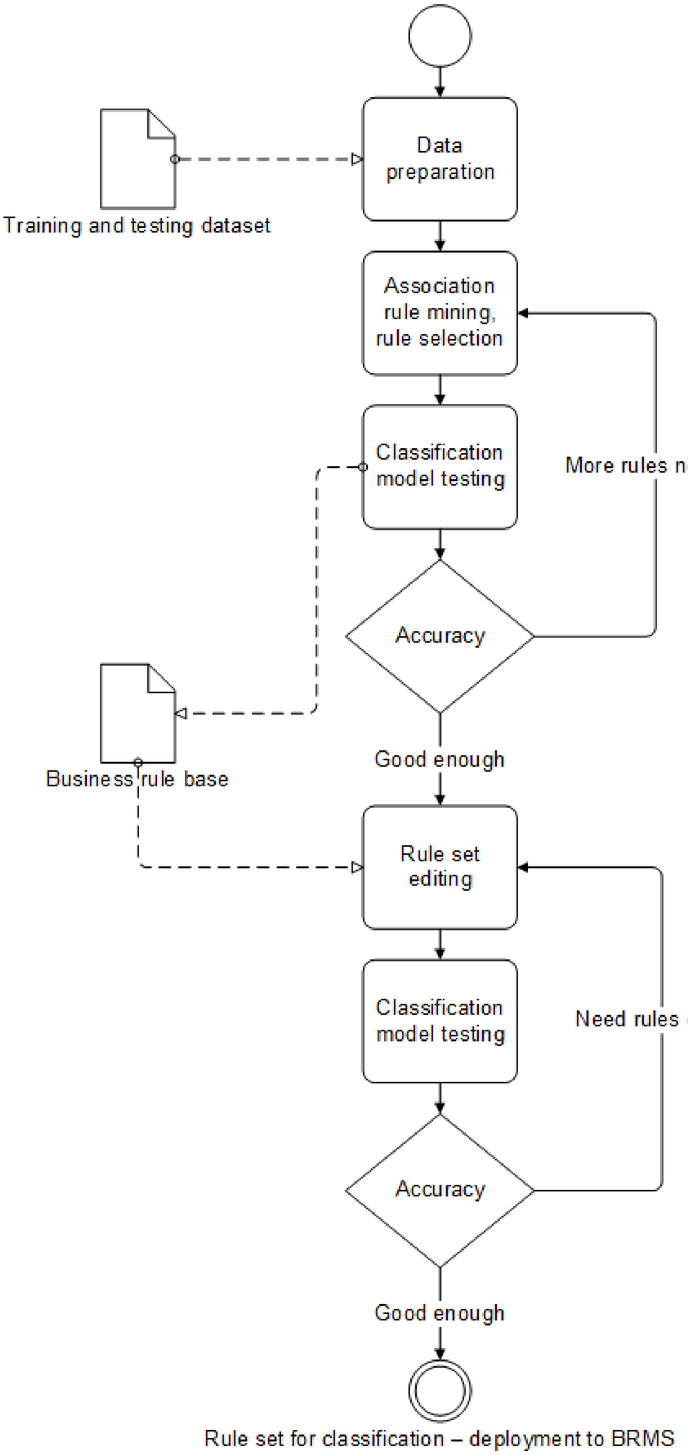

Data preparation
Training and testing dataset
Association rule mining, rule selection
Classification model testing
More rules needed
Accuracy
Business rule base
Good enough
Rule set editing
Classification model testing
Need rules editing
Accuracy
Good enough
Rule set for classification – deployment to BRMS

Fig. 1. Rule base preparation for the classification task

3.2 Association rule learning

As a rule learning algorithm, the current version uses the GUHA procedure ASSOC implemented in LISp-Miner system [6].

To start association rule learning in EasyMiner[6], the user is required to input a rule pattern definition. The user selects attributes from *attributes palette* and places them into a *rule pattern* pane, which defines the pattern that the discovered rules must match. By default, the attributes are connected with a conjunction, but the system supports also disjunction and negation connectives.

For each attribute in the rule pattern it is also possible to define the set of its values considered during mining. Supported options are *fixed value* (the user selects one value or value group defined in the preprocessing phase), *simple wildcard* (the system tries all possible values) and *dynamic binning wildcard* (the system creates new bins by merging those created during preprocessing). [7]

Additionally, the user needs to select at least one interest measure. The system supports the standard *confidence*, *support*, in addition to several other measures.

3.3 Rule selection

After association rule learning has been executed, the user is presented with a list of rules that have been discovered in the data. From this list, the user can select association rules and save them into the *Rule clipboard*. The user can then change the parameters of the rule learning task, typically by making a change in the rule pattern, and re-execute the learning. In this way, the rule clipboard accumulates rules discovered during multiple data mining tasks. The rules are saved grouped by data mining tasks.

For classification, it is vital that the rules that enter the clipboard are chosen so as to maximize accuracy on the data. To support the user in creating the best mix of rules, the system offers a "Check model" link, which displays the accuracy of the rule set.

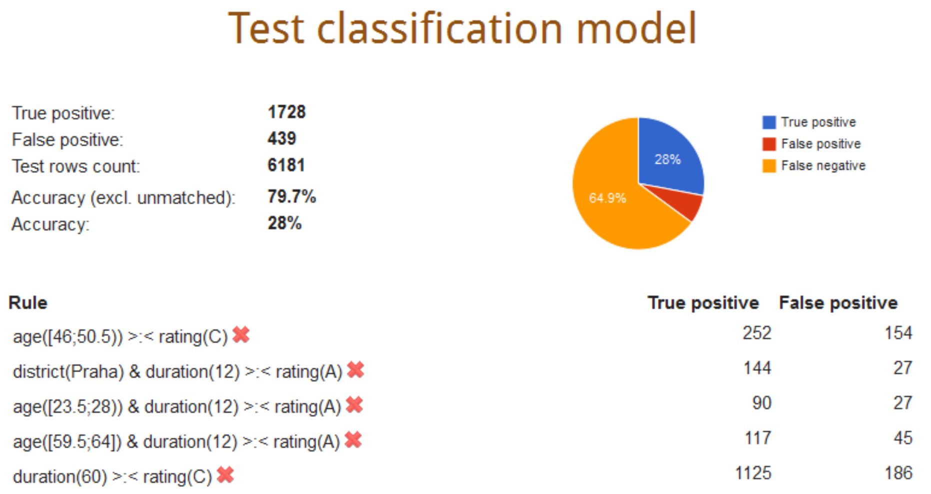
The accuracy is computed either on the training dataset or on a separate test dataset (uploaded as a CSV file) depending on the user's choice. Upon clicking on the check model link, the system exports the rules saved in the Rule Clipboard to the *Model Tester* component. Model Tester converts the rules into business rules form (specifically DRL) and uses the *Drools Expert*[8] inference engine to perform the classification.

In case of a rule conflict, which occurs if one object is matched by multiple rules, the conflict resolution strategy is as follows: matching rules are sorted according to their confidence, support and antecedent length (in this order). Out of the matching rules, only the highest ranked one is used for classification.

[6] `http://easyminer.eu`

[7] Dynamic binning is called "coefficient" setup in the LISp-Miner system, and is described in greater detail e.g. in [6].

[8] `http://drools.jboss.org`



Test classification model

True positive:	**1728**
False positive:	**439**
Test rows count:	**6181**
Accuracy (excl. unmatched):	**79.7%**
Accuracy:	**28%**

Rule	**True positive**	**False positive**
age([46;50.5)) >:< rating(C) ✖	252	154
district(Praha) & duration(12) >:< rating(A) ✖	144	27
age([23.5;28)) & duration(12) >:< rating(A) ✖	90	27
age([59.5;64]) & duration(12) >:< rating(A) ✖	117	45
duration(60) >:< rating(C) ✖	1125	186

Fig. 2. Model quality assessment

The conversion to the DRL format is described in detail in [7]. The details on the use of Drools for classification can be found in [3].

The result from the Model Tester is presented to the user in a graphical form (Figure 2). The user can see the counts of rows classified as *true positive* and *false positive*, as well as *accuracy* and *accuracy (excl. unmatched)* . The difference between the latter two statistics is that *accuracy (excl. unmatched)* does not consider objects that were not matched by any rule as misclassified, while *accuracy* does. To aid the user in optimizing the rule base, the system shows also detailed results for each rule from the Rule Clipboard.

3.4 Rule Editor

For classification model building, it is suitable to combine results (association rules) selected from more than one data mining task. The user can export rules from the Rule Clipboard into the *"business rules" base*. For practical purposes, the rules are internally saved in an XML format. The system allows to convert the rules into the *Drools DRL* format.

The user can display the rules saved in the "business rules" knowledge base and edit them using a web interface. We developed a simple rule editor, which presents the rules in a user-friendly, SBVR-like,[9] textual form (see Listing 1.1).

[9] Semantics of Business Vocabulary and Rules - `http://www.omg.org/spec/SBVR/`

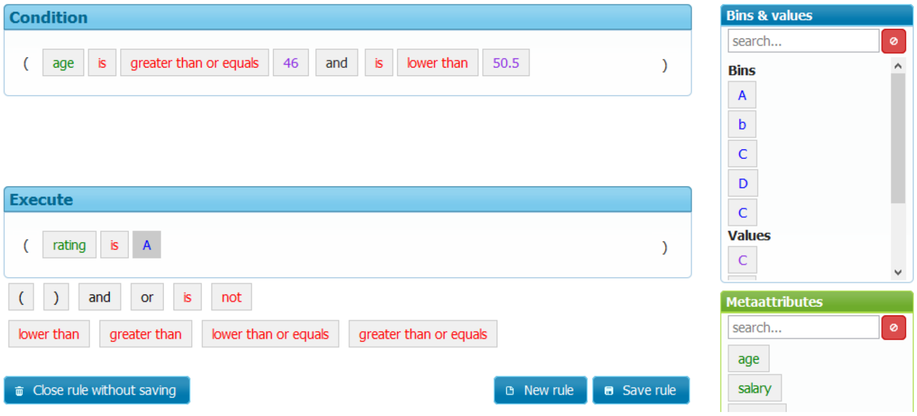
Listing 1.1. Format of rules supported in BR Editor

```
When age is in [46;50.5) then rating is C.

When age is greater than or equals 46
        and is lower than 50.5 then rating is C.
```

The rule editor (Fig. 3) allows the user to edit the rules exported from the rule clipboard, delete them, or add new rules.

Finally, the business rules base can be used for classification. The classification engine, available as a REST service, is based on Drools.



Condition

(age is greater than or equals 46 and is lower than 50.5)

Execute

(rating is A)

Fig. 3. Simple business rules editor

4 Software components

EasyMiner (easyminer.eu) is a web-based data mining system [8]. The web interface supports definition of preprocessing, rule pattern definition, results selection etc. The system is written in PHP, user interface is based on HTML5 and the MooTools JavaScript library.

LISp-Miner (lispminer.vse.cz) is used for rule learning. It is a Windows desktop application written in C++. The EasyMiner LMConnect module, implemented in C#, exposes LISp-Miner via REST web services. The communication is performed in an extension of PMML [4].

Model tester is written in Java EE. The system uses *Drools Expert* classes and the functionality is accessible through REST web services in XML or JSON. The system creates a brCBA classifier [3] from rules discovered by LISp-Miner.

Business rules editor is implemented as a JavaScript application, which uses jQuery JavaScript library and HTML5, with PHP backend.

5 Demo, further work

The demo of EasyMiner-BR, including screencasts, is accessible at `http://easyminer.eu/`. We are currently finishing the two-way integration of Business rules editor with EasyMiner system. The goal of this integration is to present the user-defined rules along the mined rules in Rule clipboard, which would allow a more informed selection of rules to the clipboard.

In the future work, we would like to explore the possibility of using the business rule knowledge base already in the process of data mining of association rules, e.g. as investigated at [2]. Another focus is the extension of the system that would allow to mix rules mined from multiple datasets in one knowledge base. Finally, we consider supporting alternative rule learning backends.

Acknowledgment

This work is supported by the University of Economics in Prague by grants no. IGA 20/2013 and institutional support (IP 400040).

References

1. Bart Goethals, Sandy Moens, and Jilles Vreeken. MIME: A framework for interactive visual pattern mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 757–760, New York, NY, USA, 2011. ACM.
2. Tomáš Kliegr, Andrej Hazucha, and Tomáš Marek. Instant feedback on discovered association rules with PMML-based query-by-example. In Sebastian Rudolph and Claudio Gutierrez, editors, *Web Reasoning and Rule Systems*, volume 6902 of *Lecture Notes in Computer Science*, pages 257–262. Springer Berlin Heidelberg, 2011.
3. Tomáš Kliegr, Jaroslav Kuchař, Davide Sottara, and Stanislav Vojíř. Learning business rules with association rule classifiers. In *RuleML14 – The 8th International Web Rule Symposium*. Springer, 2014.
4. Tomáš Kliegr and Jan Rauch. An XML format for association rule models based on GUHA method. In *RuleML-2010, 4th International Web Rule Symposium*, Berlin, Heidelberg, 2010. Springer-Verlag.
5. OpenRules, Inc. Rule learner. `http://www.openrules.com/rulelearner.htm`. Accessed: 2014-04-15.
6. Jan Rauch and Milan Šimůnek. An alternative approach to mining association rules. In *Foundations of Data Mining and Knowledge Discovery*, pages 211–231. Springer, 2005.
7. Stanislav Vojíř, Tomáš Kliegr, Andrej Hazucha, Radek Škrabal, and Milan Šimůnek. Transforming association rules to business rules: Easyminer meets Drools. In Paul Fodor, Dumitru Roman, Darko Anicic, Adam Wyner, Monica Palmirani, Davide Sottara, and François Lévy, editors, *RuleML (2)*, volume 1004 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.
8. Radek Škrabal, Milan Šimůnek, Stanislav Vojíř, Andrej Hazucha, Tomáš Marek, David Chudán, and Tomáš Kliegr. Association rule mining following the web search paradigm. In Pete rA. Flach, Tijl Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7524 of *Lecture Notes in Computer Science*, pages 808–811. Springer Berlin Heidelberg, 2012.