J. Hlaváčová (Ed.): ITAT 2017 Proceedings, pp. 235–239
CEUR Workshop Proceedings Vol. 1885, ISSN 1613-0073, © 2017 T. Kliegr, J. Kuchař, S. Vojíř, V. Zeman

EasyMiner – Short History of Research and Current Development

Tomáš Kliegr[1], Jaroslav Kuchař[2], Stanislav Vojíř[1], and Václav Zeman[1]

[1] Department of Information and Knowledge Engineering, Faculty of Informatics and Statistics, University of Economics, Prague,
W. Churchill Sq. 4, Prague 3, Czech Republic
[2] Web Intelligence Research Group, Faculty of Information Technology, Czech Technical University, Thákurova 9, 160 00, Prague 6,
Czech Republic
first.last@{vse|fit.cvut}.cz

Abstract: EasyMiner (easyminer.eu) is an academic data mining project providing data mining of association rules, building of classification models based on association rules and outlier detection based on frequent pattern mining. It differs from other data mining systems by adapting the "web search" paradigm. It is web-based, providing both a REST API and a user interface, and puts emphasis on interactivity, simplicity of user interface and immediate response. This paper will give an overview of research related to the EasyMiner project.

1 Introduction

In this paper, we present the history of research and development of the EasyMiner project http://easyminer.eu. EasyMiner is an academic data mining project providing data mining of association rules, building of classification models based on association rules and outlier detection based on frequent pattern mining.

EasyMiner was to our knowledge the first interactive web-based data mining system that supported the complete machine learning process. While today there are several web-based machine learning systems on the market[1], owing to continuous development EasyMiner provides distinct user experience. While most existing machine learning systems offer versatile user interfaces, where the user has to in some way for each task compose a new machine learning workflow, in EasyMiner the user interface is crafted to provide the "web search" experience. The user visually constructs a query against the data, and the system responds with a set of interesting patterns (presented as rules) or a classifier (Figure 1).

Over the years of development, EasyMiner served as a testbed for a number of new technologies and research ideas. The purpose of this paper is to give a brief overview of this research.

This paper is organized as follows. Section 2 is focused on SEWEBAR-CMS, the predecessor of EasyMiner, used in research on the use of domain knowledge in data mining. Section 3 focuses on association rule discovery. Section 4 presents the adaptation of EasyMiner for learning business rules and Section 5 consequently for association rule classification. Section 6 presents the current focus on outlier detection. The architecture of the system is presented in Section 7. Since the beginnings, the research was accompanied with standardization efforts, which are presented in Section 8. The current development efforts focus also on distributed computation platforms – this is covered in Section 9. Section 10 provides an overview of the features that were at some point in time developed as well as of those that are supported by the current version of EasyMiner. Finally, the conclusions present a case for using EasyMiner as a component in new project requiring data mining functionality and refers the interested reader to other publications regarding comparison with other machine learning as a service (MLaaS) systems.

2 Handling of Domain Knowledge

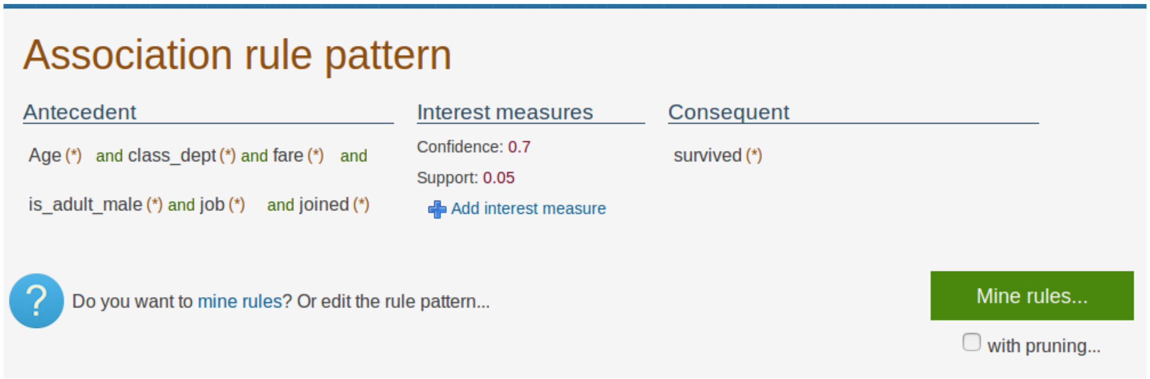
EasyMiner evolved from the SEWEBAR (SEmantic-WEB Analytical Reports) project, which focused on semantically readable machine learning. In [9], we presented SEWEBAR-CMS as a set of extensions for the Joomla! content management system (CMS) that extends it with functionality required to serve as a communication platform between the data analyst, domain expert and the report user. The system later supported elicitation of domain knowledge from the analyst [12]. Association rules discovered from data with the LISp-Miner system (http://lispminer.vse.cz) were stored in a semantic form in the SEWEBAR-CMS system. The background knowledge was used to help answer user search queries, for example, to find rules that are contradicting existing domain knowledge [6]. Another novel element in the system was the use of ontology for representation of the data mining domain.

Related research focused on improving semantic capabilities of content management systems [3] and on designing ontologies and schemata for representation of background knowledge [8, 11].

3 Association Rule Discovery

In its first release, EasyMiner provided a web-based interface for the LISp-Miner system, which was used for association rule mining [23]. EasyMiner interacted with LISp-Miner using its LM-Connect component, which is a web application providing the functionality of LISp-Miner through REST API.

[1] Such as BigML.com or Microsoft Azure.



Association rule pattern

Antecedent

Age (*) and class_dept (*) and fare (*) and is_adult_male (*) and job (*) and joined (*)

Interest measures

Confidence: 0.7

Support: 0.05

Consequent

survived (*)

Do you want to mine rules? Or edit the rule pattern...

Figure 1: Visual query designer in EasyMiner.

Table 1: Features supported in EasyMiner 2.4. Year - when was the paper describing the feature published, API - feature available in the REST API, UI - feature available in the user interface.

Feature	Year	API	UI
Content Management System [9]	2009	No	No
Semantic search over discovered rules [3]	2010	No	No
Support for GUHA extension of PMML [10]	2010	Yes	Yes
Query for related (confirming, contradicting) rules to the selected rule [6]	2011	No	No
Editor of background knowledge [12]	2011	No	No
LISp-Miner interface (disjunctions, negations, partial cedents, quantifiers, cuts, coefficients) [23]	2012	No	No
Export of business rules to Drools [21]	2013	No	No
Rule pruning with CBA [5]	2014	Yes	Yes
Evaluation of quality of classification models [20]	2014	Yes	No
Rule selection and editing for classification model building [20]	2014	Yes	No
R interface (arules package) [22]	2015	Yes	Yes
Spark backend [25]	2016	Yes	Yes
Discretization algorithms [25]	2016	Yes	No
Support for the input RDF data format	2017	Yes	No
Outlier detection [19]	2017	Yes	No

EasyMiner with LISp-Miner backend offered several unique features: 1. negation on attributes, 2. disjunction between attributes, 3. subpatterns allowing for scoping logical connectives, 4. multiple interest measures (called quantifiers in GUHA), 5. mines directly on multivalued attributes, no need to create "items", 6. dynamic binning operators (called coefficients in GUHA), 7. PMML-based import and export, 8. grid support.

Since LM-Connect component is no longer developed and maintained, the integration of the current version of EasyMiner and LISp-Miner is thus currently not working.[2]

The current version of EasyMiner primarily relies on the R arules package [2], which wraps a C implementation of the apriori association rule mining algorithm [1].

4 Learning Business Rules

One of the first use cases for EasyMiner was learning business rules. In [21] we presented a software module for EasyMiner, which allows to export selected rules to Business Rules Management System (BRMS) Drools, transforming the output of association rule learning into the DRL format supported by Drools. We found that the main obstacles for a straightforward use of association rules as candidate business rules are the excessive number of rules discovered even on small datasets, and the fact that contradicting rules are generated. In [5] we propose that a potential solution to these problems is provided by the seminal association rule classification algorithm CBA [16]. In [20] we presented a software module for EasyMiner, which allows the domain expert to edit the discovered rules.

5 Association Rule Based Classification

In [5] we started to use the CBA algorithm for postprocessing association rule learning results into a classifier. In [22] we presented an extension for EasyMiner for building of classification models. A benchmark against standard symbolic classification algorithms on a news recommender task was presented in [7].

[2] It should be noted that all the features list above can be used directly from the LISp-Miner system.

6 Outlier Detection

The most recent addition of new tasks supported by EasyMiner is frequent pattern-based anomaly (outlier) detection. The main idea of the approach is that if an instance contains more frequent patterns, it is unlikely to be an anomaly. The presence or absence of the frequent patterns is then used to assign the deviation level [4]. In [19] we present extension of EasyMiner REST API with our innovated outlier detection algorithm called *Frequent Pattern Isolation* (FPI)[15] that is inspired by an existing algorithm called Isolation Forests (IF) [17, 18]. Since PMML does not yet support outlier (anomaly) detection, in [14] we present our proposal for a new PMML outlier model. The goal of our work was to design modular solution that would support broader range of anomaly detection algorithms including our FPI method.

7 EasyMiner Architecture

During the development of EasyMiner system, its architecture was transformed to multiple reusable web services. A schema of the architecture is shown in Figure 2. All the services are fully documented in Swagger.

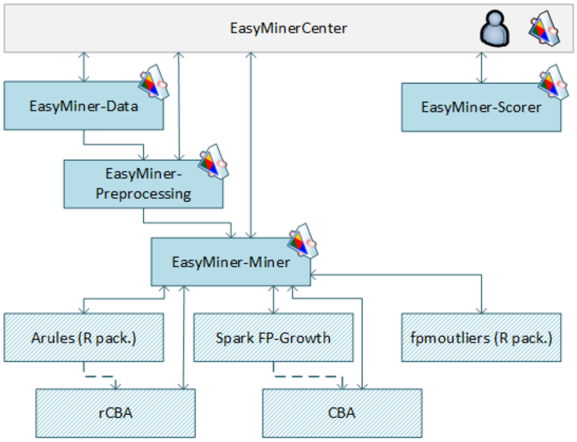


Figure 2: Architecture of the system EasyMiner

The central component (service) is **EasyMinerCenter**. This component integrates the functionality of other services and provides the main graphical web interface and REST API for end users. Internally, this component provides user account and task management, stores discovered association rules and works as authentication service for other components.

For storing and preparing data before mining, the system uses services EasyMiner-Data and EasyMiner-Preprocessing. **EasyMiner-Data** is a web services for management of data sources. It supports upload of data files in CSV and RDF and stores them into databases as the set of transactions. **EasyMiner-Preprocessing** service supports creation of datasets from data sources stored using EasyMiner-Data using user-defined preprocessing methods. The attributes for data mining are created from uploaded data fields using one of these preprocessing algorithms: each value-one bin, enumeration of intervals, enumeration of nominal values, equidistant intervals, equifrequent intervals, equisized intervals (by minimal support of every interval). The preprocessing algorithms as well as data storage are independent of the selected data mining algorithm. The implemented web services support hashing functionality to avoid potentially problems with special characters in attribute names and its values. The mining following services work on the "safe" datasets with hashed values.

The main data mining functionality is provided by the service **EasyMiner-Miner**. This web service provides association rule learning, prunning of discovered association rule sets and building of classification models and outlier detection. EasyMiner-Miner initializes execution of used R packages and another algorithms.

EasyMiner-Scorer is a web service for testing of classification models based on association rules.

8 Distributed Backend: Spark/Hadoop

As laid out in the previous section, EasyMiner is modular in terms of mining backends. In addition to the default mining backend provided by the arules and rCBA packages, EasyMiner supports an alternate one built on top of Apache Spark/Hadoop introduced in [25].

The Spark backend is suitable for larger datasets, which can benefit from parallel computation distributed over multiple machines. The Spark backend also uses FP-Growth frequent pattern mining algorithm instead of apriori. FP-Growth is generally considered as faster than apriori. However, for smaller datasets using apriori with the R backend is recommended as it provides faster response times, due to the ability of the implementation to provide intermediate results as the mining progresses.

9 Standardization Efforts (PMML)

Already the earliest research related to EasyMiner was linked to work on standardization efforts. While association rules were supported already in the early versions of PMML, the industry standard format for exchange of data mining models, the GUHA method that was initially used did not comply to this standard, since it produced rules containing number of constructs not supported by PMML. Since our research involved background knowledge elicited from domain experts, definition of data format supporting this type of knowledge was also required.

In [8] we proposed a topic map-based ontology for association rule learning, which was based on the GUHA method and in [11] an extension of this approach that dealt with domain knowledge. An extension of PMML for

GUHA-based models was presented in [10] and for handling of background knowledge [13]. Neither of these efforts was successful – the ISO Topic Maps standard waded in favour of the W3C RDF/OWL stack. The industry was not concerned with exchange of background knowledge at the time, and support of GUHA method, implemented essentially only by the LISp-Miner system, increased complexity of the models as opposed to the existing PMML association rule models.[3] Our latest standardization effort is related to outlier detection [14] and targets PMML. This proposal is closes industry adoption as it was included into a roadmap for the next release of PMML.

10 Features in the EasyMiner Version 2.4

Table 1 presents an overview of the most salient features that were in some for published between 2009, when the first paper on EasyMiner's predecessor SEWEBAR-CMS appeared, and 2017, when the current version of EasyMiner was released. As follows from the table, a number of features is not supported in the current release.

11 Using EasyMiner in Your Project

During the years of development, EasyMiner was extensively used by over thousand of students at the Faculty of Informatics and Statistics to complete their assignments in association rule learning. The software has also been used in several applied research projects. For example, within the `linkedtv.eu` project EasyMiner was used to analyze user preferences and within the `openbudgets.eu` project to analyze budgetary data.

The full project is based on composition of components and services with fully documented REST APIs. Most of the components and services[4] are available under open source Apache License, Version 2.0. This is an important factor which differentiates EasyMiner from the commercial MLaaS offerings. For a more detailed comparison with other machine learning systems refer to [24].

In addition to the visual web-based interface, the project exposes a REST API. This API provides full functionality of EasyMiner, including also functions, which are not yet available in the GUI. It is possible to use this API to extend your own project by data mining functionality. It is suitable for building of mashup applications or data processing using script languages. An example of data mining using API is available at `http://www.easyminer.eu/api-tutorial`.

EasyMiner can also be extended with new algorithms - rule mining, outlier detection or scorer service. For this purpose, the integration component EasyMinerCenter provides documented interfaces in PHP.

[3] Currently, EasyMiner supports export of association rule models in formats GUHA PMML also as in standard form PMML 4.3 Association Rules.

[4] The main services were presented in section 7.

Acknowledgment

This paper was supported by IGA grant 29/2016 of the University of Economics, Prague.

References

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216. ACM Press, 1993.

[2] Michael Hahsler, Sudheer Chelluboina, Kurt Hornik, and Christian Buchta. The arules r-package ecosystem: analyzing interesting patterns from large transaction data sets. *Journal of Machine Learning Research*, 12(Jun):2021–2025, 2011.

[3] Andrej Hazucha, Jakub Balhar, and Tomáš Kliegr. A PHP library for Ontopia-CMS integration. In *TMRA 2010*. University of Leipzig, 2010.

[4] Z. He, X. Xu, Z. Huang, and S. Deng. FP-outlier: Frequent pattern based outlier detection. *Computer Science and Information Systems/ComSIS*, 2(1):103–118, 2005.

[5] Tomáš Kliegr, Jaroslav Kuchař, Davide Sottara, and Stanislav Vojíř. Learning business rules with association rule classifiers. In Antonis Bikakis, Paul Fodor, and Dumitru Roman, editors, *Rules on the Web. From Theory to Applications: 8th International Symposium, RuleML 2014, Co-located with the 21st European Conference on Artificial Intelligence, ECAI 2014, Prague, Czech Republic, August 18-20, 2014. Proceedings*, pages 236–250, Cham, 2014. Springer International Publishing.

[6] Tomáš Kliegr, Andrej Hazucha, and Tomáš Marek. Instant feedback on discovered association rules with PMML-based query-by-example. In *Web Reasoning and Rule Systems*. Springer, 2011.

[7] Tomáš Kliegr and Jaroslav Kuchař. Benchmark of rule-based classifiers in the news recommendation task. In Josiane Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth J. F. Jones, Eric SanJuan, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, volume 9283 of *Lecture Notes in Computer Science*, pages 130–141. Springer, 2015.

[8] Tomáš Kliegr, Marek Ovečka, and Jan Zemánek. Topic maps for association rule mining. In *Proceedings of TMRA 2009*. University of Leipzig, 2009.

[9] Tomáš Kliegr, Martin Ralbovský, Vojtěch Svátek, Milan Šimunek, Vojtěch Jirkovský, Jan Nemrava, and Jan Zemánek. Semantic analytical reports: A framework for post-processing data mining results. In *ISMIS'09: 18th International Symposium on Methodologies for Intelligent Systems*, pages 453–458. Springer, 2009.

[10] Tomáš Kliegr and Jan Rauch. An XML format for association rule models based on the GUHA method. In *Proceedings of the 2010 International Conference on Semantic Web Rules*, RuleML'10, pages 273–288, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] Tomáš Kliegr, Vojtěch Svátek, Milan Šimůnek, Daniel Štastný, and Andrej Hazucha. An XML schema and a topic map ontology for formalization of background knowledge in data mining. In *IRMLeS-2010, 2nd ESWC Workshop on Inductive Reasoning and Machine Learning for the Semantic Web, Heraklion, Crete, Greece*, 2010.

[12] Tomáš Kliegr, Vojtěch Svátek, Milan Šimůnek, and Martin Ralbovský. Semantic analytical reports: A framework for post-processing of data mining results. *Journal of Intelligent Information Systems*, 37(3):371–395, 2011.

[13] Tomáš Kliegr, Stanislav Vojíř, and Jan Rauch. Background knowledge and PMML: first considerations. In *Proceedings of the 2011 workshop on Predictive markup language modeling*, PMML '11, pages 54–62, New York, NY, USA, 2011. ACM.

[14] Jaroslav Kuchař, Adam Ashenfelter, and Tomáš Kliegr. Outlier (anomaly) detection modelling in PMML. In *RuleML 2017 Poster and Challenge Proceedings*. CEUR-WS, 2017.

[15] Jaroslav Kuchař and Vojtěch Svátek. Spotlighting anomalies using frequent patterns. In *KDD 2017 Workshop on Anomaly Detection in Finance, Halifax, Nova Scotia, Canada*, 2017.

[16] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, pages 80–86. AAAI Press, 1998.

[17] F. T. Liu, K. M. Ting, and Z. H. Zhou. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08)*, pages 413–422, 2008.

[18] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data*, 6(1):3:1–3:39, March 2012.

[19] Stanislav Vojíř, Jaroslav Kuchař, Václav Zeman, and Tomáš Kliegr. Using easyminer API for financial data analysis in project openbudgets.eu. In *RuleML 2017 Poster and Challenge Proceedings*. CEUR-WS, 2017. To appear.

[20] Stanislav Vojíř, Přemysl Václav Duben, and Tomáš Kliegr. Business rule learning with interactive selection of association rules. *RuleML Challenge*, 2014, 2014.

[21] Stanislav Vojíř, Tomáš Kliegr, Andrej Hazucha, Radek Skrabal, 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 Francois Lévy, editors, *RuleML-2013 Challenge*, volume 1004 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.

[22] Stanislav Vojíř, Václav Zeman, Jaroslav Kuchař, and Tomáš Kliegr. Easyminer/R preview: Towards a web interface for association rule learning and classification in R. In *Challenge+ DC@ RuleML*, 2015.

[23] Radek Škrabal, Milan Šimůnek, Stanislav Vojíř, Andrej Hazucha, Tomáš Marek, David Chudán, and Tomáš Kliegr. Association rule mining following the web search paradigm. In Peter A. 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.

[24] Václav Zeman. Analýza cloudového řešení akademického nástroje pro dolování pravidel z databází. *Systémová Integrace*, 23, 2016.

[25] Václav Zeman, Stanislav Vojíř, Jaroslav Kuchař, and Tomáš Kliegr. Využití cloudu pro dolování asociačních pravidel z velkých dat přes webové rozhraní. In *WIKT/DaZ 2016*, 2016.