Rough Inclusion Based Toy Decision Systems Generator For Presenting Data Mining Algorithms*

Piotr Artiemjew[1][0000−0001−5508−9856]

Faculty of Mathematics and Computer Science, University of Warmia and Mazury in Olsztyn, 10-710 Olsztyn, Poland artem@matman.uwm.edu.pl
http://wmii.uwm.edu.pl/ artem/

Abstract. In this work, we address researchers who want to use various meaningful, real-world data to demonstrate their algorithms using toy examples. We propose a tool for generating toy decision systems from original decision systems addressing real decision problems. The toy decision systems are generated using a concept-dependent granulation technique and feature selection using the relative discernibility matrix. The resulting toy decision systems represent the essence of real decision systems, they provide meaningful data that can be used to demonstrate the action of data mining algorithms.

Keywords: Decision Systems · Toy Decision Systems · Generator · Data Mining · Tiny Data · Toy Data · Small Data · Rough Sets

1 Introduction

The article is a technical note presenting a novel toy decision system generator for use in presenting data mining algorithms. We present an idea for generating toy decision systems from real-life decision systems. The outcome of this work is a working generator for open use [1]. The tool has been developed using techniques derived from the rough set theory - [2]. In particular, we used the concept-dependent granulation technique to reduce the size of decision systems, a method derived from the technique proposed by Polkowski in [4] and developed in the paper [6]. To reduce the number of attributes, we used a relative discernibility matrix. The technique used is a method developed by Polkowski and further developed by Artiemjew for reducing the volume of decision systems while preserving their internal knowledge. A very extensive study in this context is carried out in the monograph [5].

1.1 Motivation

Probably every data scientist has come across Quinlan's toy decision system [7] (the toy decision problem of playing tennis), which has become a popular deci-

* This work has been supported by the grant from Ministry of Science and Higher Education of the Republic of Poland under the project number 23.610.007-000

sion system for presenting toy examples of ideas for new data science algorithms. Quinlan used this decision system to demonstrate the ID3 technique, the foundation for the C4.5 method [7]. The idea to build an application for generating small meaningful decision systems comes from the fact that there is a lack of such a tool on the Internet. The author has built it for his own use and wants to share its functionalities with other researchers.

The rest of this work consists of the following sections. In section 2 we have a detailed description of the methodology used. In section 3 we have a demonstration of the generator. In section 4 we have a summary of publications and future plans.

2 Methodology

A general scheme for creating small decision systems is to granulate the original data and to reduce dimensions. We will first introduce the method of reducing the number of objects of decision systems.

2.1 Granulation Techniques - Reduction of the Number of Objects

Our methods are based on rough inclusions. Introduction to rough inclusions in the framework of rough mereology is available in Polkowski [3] – [5]. We start with a detailed description of the basic method - see [4] - standard granulation. Let us consider the decision system (U, A, d), where U is the universe of objects, A the set of condition attributes, $d \notin A$ is the decision attribute, and r_{gran} granulation radius from the set $\{0, \frac{1}{|A|}, \frac{2}{|A|}, ..., 1\}$. The standard rough inclusion relation μ, for $u, v \in U$ and for selected r_{gran} is defined as

$$\mu(v, u, r_{gran}) \Leftrightarrow \frac{|IND(u,v)|}{|A|} \geq r_{gran}, \text{ where } IND(u,v) = \{a \in A : a(u) = a(v)\}, \tag{1}$$

For each object $u \in U$, and selected r_{gran}, we compute the *standard granule* $g_{r_{gran}}(u)$ as follows, $g_{r_{gran}}(u)$ is $\{v \in U : \mu(v, u, r_{gran})\}$. In the next step, we use selected strategy to cover the training decision system U by computed granules - the random choice is the simplest among the most effective studied in [5]). All methods being studied are available in [5] (pages 105 – 220). And in the last step, granular reflection of training set is computed with the use of Majority Voting procedure. The ties are resolved randomly. A concept–dependent (cd) granule $g^{cd}_{r_{gran}}(u)$ of the radius r_{gran} of u is defined as follows:

$$v \in g^{cd}_{r_{gran}}(u) \text{ if and only if } \mu(v, u, r_{gran}) \text{ and } (d(u) = d(v)) \tag{2}$$

2.2 Attribute Selection Using a Relative Discernibility Matrix

We used the following tool to select the attributes of the original decision system. For the decision system (U, A, d), where $A = \{a_1, a_2, ..., a_k\}$, $a_i \in A$ we define matrix grids, for pairs of objects $u_{j_1}, u_{j_2} \in U$.

$$c_{j_1 j_2}^{a_i, relative} = \begin{cases} 1, \ if \ a_i(u_{j_1}) ! = a_i(u_{j_2}), d(u_{j_1}) ! = d(u_{j_2}), j_1 < j_2 \\ 0, \ otherwise \end{cases} \tag{3}$$

$class\_separation\_level_{a_i} = \sum_{u_{j_1}, u_{j_2} \in U} c_{j_1 j_2}^{a_i, relative}$

After calculating the *class_separation_level* for all condition attributes, we rank them and select a fixed number for the final decision system. The higher the value of the *class_separation_level* parameter, the better the class separation.

3 Demostration of the Generator

To demonstrate the operation of our tool, we selected several decision systems from the UCI repository [8]. We implemented our tool in django technology. The detailed appearance of our application available at [1], can be seen in Figure 1. The application is divided into three parts - input, intermediate and output layer. The left window shows information about sample data and recommended parameters. The second window is for entering parameters. In the field *Link to your decision system* we have space to paste a link to the data, it can be in csv (comma-separated values) or tsv (tab-separated values) format. In the *Class size of the data drawn* field we can specify the number of objects of particular classes to be drawn from the original decision system. This step allows us to process large decision systems. The *No. of attributes to extract*? field specifies the number of attributes to be extracted. In the field *Degree of approximation* is the number of the degree of approximation, which is from the set $\{0, 1, \ldots, |A|\}$. The last window is used to display the resulting decision system.

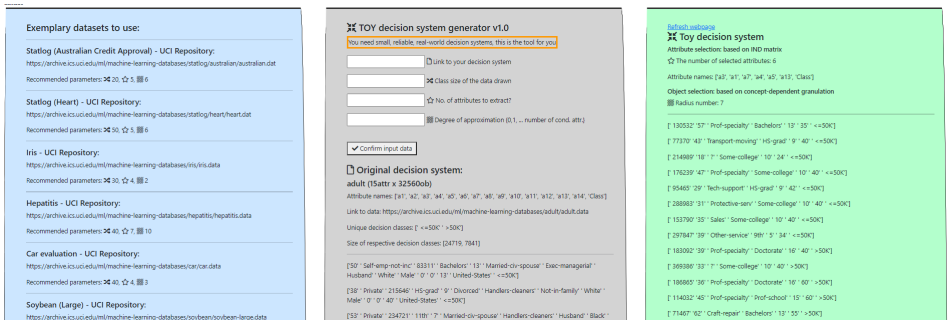

Fig. 1. Application screenshot. The middle window is used for data entry, the left window is information about the decision systems that have been prepared to test the application, the right window is the resulting window with the extracted toy decision system.

In Tab. 1, one can see an example of the effect of our generator - toy decision systems generated based on the Iris and Car ([8]) data sets.

Table 1. An example of toy decision systems generated with our application.

From the Iris data set

a3	a4	a1	Class
1.5	0.1	5.2	Iris-setosa
4.2	1.5	5.9	Iris-versicolor
5.8	2.2	6.5	Iris-virginica

From the Car data set.

a3	a2	a1	a5	Class
4	med	vhigh	big	acc
5more	high	low	big	acc
4	med	high	small	acc
2	low	med	big	good
3	med	low	small	good
4	low	med	med	unacc
4	vhigh	vhigh	big	unacc
5more	high	low	med	unacc
2	high	low	big	vgood
4	low	low	big	vgood
4	low	med	med	vgood

4 Conclusions

The current paper presents a tool for creating toy decision systems from large real-world data. To achieve this goal, a decision system approximation technique - the concept-dependent granulation method - and a feature selection method based on a relative discernibility matrix are used. This tool is dedicated to researchers who want to present their data science algorithms on small meaningful data. The version we present is dedicated to symbolic data. The tool works on sample recommended parameters. In the future, we plan to extend the tool to generate decision systems for data of any type. Additionally, a desktop version using tkinter technology is being developed.

References

1. Toy decision system generator, http://toyds.herokuapp.com/generator/v1/. Last accessed 12 Apr 2022
2. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341—356 (1982). https://doi.org/10.1007/BF01001956
3. Polkowski, L.: Formal granular calculi based on rough inclusions, In: Proceedings of IEEE 2005 Conference on Granular Computing GrC05, pp. 57–62. IEEE Press, Beijing, China (2005)
4. Polkowski, L.: A model of granular computing with applications, In: Proceedings of IEEE 2006 Conference on Granular Computing GrC06, pp. 9–16. IEEE Press, Atlanta, USA (2006)
5. Polkowski, L., Artiemjew, P.: Granular Computing in Decision Approximation - An Application of Rough Mereology, In: Intelligent Systems Reference Library 77, Springer, ISBN 978-3-319-12879-5, pp. 1–422 (2015).
6. Artiemjew, P.: Classifiers from Granulated Data Sets: Concept Dependent and Layered Granulation, In: Proceedings RSKD'07. The Workshops at ECML/PKDD'07, pp. 1–9., Warsaw Univ. Press, Warsaw (2007)
7. Quinlan, J., R.: C4.5: Programs for Machine Learning, Morgan Kaufmann, Kluwer Academic Publishers (1993). https://doi.org/10.1023/A:1022645310020
8. UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/index.php. Last accessed 12 Apr 2022