Rule Extraction via Dynamic Discretization with an Application to Air Quality Modelling*

J. Kamińska[1 [0000−0002−0157−516X]], E. Lucena-Sánchez[2,3 [0000−0001−9312−1175]], G. Sciavicco[4 [0000−0002−9221−879X]], and I.E. Stan[5,6 [0000−0001−9260−102X]]

[1] Dept. of Math., Wrocław University of Environmental and Life Sciences
joanna.kaminska@upwr.edu.pl
[2] Dept. of Phy., Inf., and Math., University of Modena e Reggio Emilia
[3] Dept. of Math. and Comp. Sci., University of Ferrara
estrella.lucenasanchez@unife.it
[4] Dept. of Math. and Comp. Sci., University of Ferrara
guido.sciavicco@unife.it
[5] Dept. of Math., Phy., and Comp. Sci., University of Parma
[6] Dept. of Math. and Comp. Sci., University of Ferrara
ioneleduard.stan@unife.it

Abstract. Association rule extraction is a very well-known and important problem in machine learning, and especially in the sub-field of explainable machine learning. Association rules are naturally extracted from data sets with Boolean (or at least categorical) attributes. In order for rule extraction algorithms to be applicable to data sets with numerical attributes as well, data must be suitably discretized, and a great amount of work has been devoted to finding good discretization algorithms, taking into account that optimal discretization is a NP-hard problem. Motivated by a specific application, in this paper we provide a novel discretization algorithm defined as an (heuristic) optimization problem and solved by an evolutionary algorithm, and we test its performances against well-known available solutions, proving (experimentally) that we are able to extract more rules in a easier way.

Keywords: Association rule extraction · Optimization problem · Evolutionary algorithm.

1 Introduction

Among the most relevant problems in machine learning there are *classification problem* and *association rule extraction problem*. An *association rule* is an if-then statement that helps to show the probability of certain relationship among the

* This research has been partially supported by the Italian INDAM GNCS project *Strategic Reasoning and Automatic Synthesis of Multi-Agent Systems*

attributes of a data set. While classification is a *global* problem (e.g., a logistic regression model that classifies all instances), association rule extraction is a *local* problem, that makes it possible for unusual, uncommon relationships to emerge. The first, and most successful, rule extraction algorithm is Agrawal and Srikant's APRIORI [1], which is originally designed to extract, with a deterministic approach, meaningful rules from a data set of Boolean attributes. APRIORI has been later generalized to deal with numerical attributes (approaching, in a sense, the discretization problem) in [3], and to extract temporal rules (on a very limited basis) in [2]. The quality of association rules is generally measured by their *support* (that is, relevance within the data set) and their *confidence* (that is, precision); depending on the particular application, though, other, more specific quality measures can be introduced.

The approaches to the discretization (or binning) problem, or the problem of mining *quantitative* association rules can be categorized into deterministic (greedy), or non-deterministic (usually based on metaheuristics). Deterministic *static* approaches simply cut the domain of a continuous attribute into a predefined number of bins before the mining process. Agrawal's solution to discretization can be considered static. Indeed, it is built on a deterministic procedure that partitions each quantitative attribute one-by-one (i.e., without considering the distribution of the other attributes) into equal-width intervals, and on a modification to the original APRIORI algorithm to handle sets of items that are generalizations of others (e.g., saying that the age of someone varies in the interval $[20, 29]$ is more general than saying that it varies in the interval $[20, 24]$) and to take into account a newly introduced *interestingness measure*. Moreover, adjacent intervals can be merged to compensate the low confidence of rules in some cases. Discretization algorithms are available in some open-source learning frameworks that belong to this category and allow both equal-width and equal-frequency discretizations, but are separated from the rule learning phase, which is applied to the resulting, discretized, data set: WEKA's discretization filter [22], Python's k-bins discretizer [18], and R's discretization library [19]. Deterministic *distance-based* discretization is a possible alternative to static methods, and it is based on clusterization; examples include [4, 13, 15, 21], among others. Non-deterministic discretization, on the other hand, is based on designing an optimization, multivariate problem [14, 20]: rules are extracted, and evaluated on a discretized data set that is progressively modified; the main drawback of such an approach is that the rule extraction algorithm must be re-designed. Finally, most of the existing approaches are general purpose, but in many cases the intended applications are represented by data sets with discrete numerical attributes (e.g., number of cars possessed, age, and so on); applying a discretization algorithm for rule extraction from data sets with continuous attributes may require a more specialized approach.

In this paper we consider a data set that represents the measurement of several air quality parameters on a specific monitoring station in the center of Wrocław (Poland) during the interval of three years. Such a data set has been studied in the context of global explanation models (e.g., in [11,12]), with inter-

esting, but possibly incomplete, results; this is usually the case with real-world data, on which one can often expect to be able to extract a general model that leaves room to particular cases. In this paper we aim to extract meaningful local rules, which may be useful to model the particular situations that occurred during the measurement period. We propose and test a multivariate, non-deterministic, distribution-based discretization algorithm that can be coupled with a pre-existing rule extraction algorithm, such as APRIORI, and implemented via an evolutionary algorithm. We prove that our solution produces more rules w.r.t. the particular situations than those produced by the discretization algorithms available in some open-source learning libraries. Finally, we prove that, like most preprocessing task, optimal discretization is NP-hard, which indicates that heuristic-based approaches, such as ours, are to be preferred; this result, shown in the Appendix, is not necessary for the understanding of the rest of the paper.

The paper is organized as follows. Section 2 presents the needed preliminaries, Section 3 formalizes the dynamic discretization process as an optimization problem together with implementation details, Section 4 discusses the data and the experimental setup, and, finally, Section 5 discusses the results of the proposed method before concluding in Section 6.

2 Background

Discretization. The discretization of the numerical attributes of a data set is a preprocessing phase, much like feature selection or outlier detection. Preprocessing approaches, which are designed to enable or improve a data mining process on a data set, are classically separated into filters, wrappers, or embedded algorithms. A *filter*, which is by large the most common type, is an algorithm that uses statistical indicators to perform the preprocessing; it is independent from the data mining or model extraction phase, which is applied to the resulting data set. A *wrapper* is a combined approach, in which the result of the preprocessing phase is continuously evaluated using the performances of a selected data mining algorithm, in search of an optimal solution. *Embedded* algorithms for preprocessing are integrated data mining algorithms, in which the preprocessing is part of the mining process; by nature, embedded algorithms are more difficult to implement, and therefore less common. The most relevant examples of discretization filters are included in some open-source learning frameworks, namely WEKA's discretization filter [22], Python's k-bins discretizer [18], and R's discretization library [19]; the differences among them are mainly related to their implementation, and they tend to behave in a very similar way. An unsupervized discretization filter usually offers two modalities, namely *equal-width* bins (that is, bins in which the endpoints of the describing interval have all the same distance), and *equal-frequency* bins (that is, bins that have roughly the same number of instances). Attribute preprocessing algorithms, such as discretization algorithms, can be also separated into *univariate* (in which attributes are considered one-by-one) and *multivariate* (in which the decisions are taken

for subsets of attributes all together); the most common filters, as those mentioned above, are all univariate. Among the proposed multivariate filters for discretization, *distance-based* discretization has been studied in [4, 13, 15, 21], among others. Since optimal preprocessing is usually a computational unfeasible problem, the proposed deterministic solutions are generally sub-optimal (greedy), and, sometimes, non-deterministic (i.e., heuristic-based) solutions are advocated; however, filters are typically deterministic. In the current literature, non-deterministic solutions for discretization are usually embedded algorithms, in which, as we have recalled, the rule-extraction phase is re-designed to include discretization, as in [3, 14, 17, 20]. Thus, summarizing, discretization filters are mostly univariate and greedy, while embedded discretization algorithms require the implementation of an ad-hoc rule extraction method.

Multiobjective optimization. A *multiobjective optimization problem* (see, e.g. [6]) can be formally defined as the optimization problem of simultaneously minimizing (or maximizing) a set of t arbitrary functions:

$$\begin{cases} \min / \max \ f_1(\bar{x}) \\ \min / \max \ f_2(\bar{x}) \\ \dots \\ \min / \max \ f_t(\bar{x}), \end{cases} \tag{1}$$

where $\bar{x}$ is a vector of decision variables. A multiobjective optimization problem can be *continuous*, in which we look for real values, or *combinatorial*, where we look for objects from a countably (in)finite set, typically integers, permutations, or graphs. Maximization and minimization problems can be reduced to each other, so that it is sufficient to consider one type only. A set $\mathcal{F}$ of solutions for a multiobjective problem is *non-dominated* (or *Pareto optimal*) if and only if for each $\bar{x} \in \mathcal{F}$, there exists no $\bar{y} \in \mathcal{F}$ such that (i) there exists i $(1 \leq i \leq t)$ that $f_i(\bar{y})$ improves over $f_i(\bar{x})$, and (ii) for every j, $(1 \leq j \leq t,\ j \neq i)$, $f_j(\bar{x})$ does not improve $f_i(\bar{y})$. In other words, a solution $\bar{x}$ *dominates* a solution $\bar{y}$ if and only if $\bar{x}$ is better than $\bar{y}$ in at least one objective, and it is not worse than $\bar{y}$ in the remaining objectives. We say that $\bar{x}$ is *non-dominated* if and only if there is no other solution that dominates it. The set of non-dominated solutions from $\mathcal{F}$ is called *Pareto front*. Optimization problems can be approached in several ways; among them, *multiobjective evolutionary algorithms* are a popular choice (see, e.g. [9, 10, 16]). Wrapper preprocessing algorithms can be easily defined as multiobjective optimization problems, as it is the case, for example, for feature selection; in this paper we show how discretization for rule extraction can follow the same pattern.

3 Dynamic Discretization

Rule extraction. For a given *categorical* data set $\mathcal{A}$, with *categorical attributes* $A = \{A_1, \dots, A_n\}$, an *association rule* ρ is an object of the type:

$$\rho : \gamma_1 \wedge \dots \wedge \gamma_s \Rightarrow \gamma_{s+1}$$

where $s < n$ and γ_i belongs to the domain of A_i, for each i. As defined by Agrawal and Srikant [1], a rule such as ρ can be evaluated by its *support*, that is, the fraction of the data set in which every instance is characterized by having both $\gamma_1, \ldots, \gamma_s$ (*antecedent*) and γ_{s+1} (*consequent*), and its *confidence*, that is, the ratio between the support and the subset of instances with $\gamma_1, \ldots, \gamma_s$. So, the greater the support the more influential is the rule in the data set, and the greater the confidence the greater is its precision; in general, we search for meaningful rules (e.g., particular situations) with a high confidence. Discretizing a numerical data set allows one to extract association rules from it, as it converts numerical attributes into categorical ones. In the following, we refer to the categories of a categorical attribute as *labels*.

Binning. Given a numerical attribute, the first step to dynamic discretization is defining a parametric function that returns the bins. Thus, let now $\mathcal{A}$ be a *numerical* data set with attributes $A = \{A_1, \ldots, A_n\}$:

$$\mathcal{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}$$

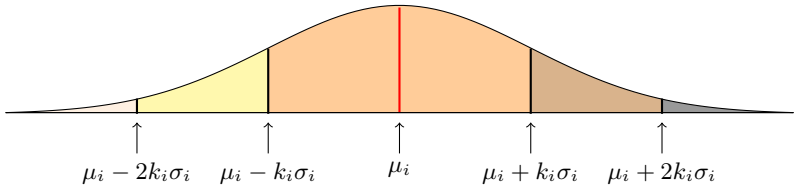
For a variable A_i, we denote by μ_i its mean in A, and by σ_i its standard deviation; moreover, let $k_i \in (0, 1]$ be a *displacement* parameter. For a given number l_i of expected bins in which we want to discretize the domain of A_i, let $\gamma_i^1, \ldots, \gamma_i^{l_i}$ be the corresponding labels. Now, let $v(A_i)$ be a certain value of the variable A_i; by writing $v(A_i) \in \gamma_i^p$ we indicate that the interval/bin denoted by the label γ_i^p includes the value $v(A_i)$. Then, we set that $v(A_i) \in \gamma_i^p$ if:

$$\begin{cases} v(A_i) < \mu_i - \lfloor l_i/2 \rfloor k_i \sigma_i & \text{if } p = 1 \\ \mu_i - (\lceil l_i/2 \rceil - p + 1) k_i \sigma_i \leq v(A_i) < \mu_i - (\lceil l_i/2 \rceil - p) k_i \sigma_i & \text{if } 1 < p < \lceil l_i/2 \rceil \\ \mu_i - k_i \sigma_i \leq v(A_i) \leq \mu_i + k_i \sigma_i & \text{if } p = \lceil l_i/2 \rceil \\ \mu_i + (p - \lceil l_i/2 \rceil) k_i \sigma_i < v(A_i) \leq \mu_i + (p - \lceil l_i/2 \rceil + 1) k_i \sigma_i & \text{if } \lceil l_i/2 \rceil < p < l_i \\ \mu_i + \lfloor l_i/2 \rfloor k_i \sigma_i < v(A_i) & \text{if } p = l_i \end{cases} \tag{2}$$

for an odd l_i, and if:

$$\begin{cases} v(A_i) \leq \mu_i - (l_i/2 - p) k_i \sigma_i & \text{if } p = 1 \\ \mu_i - (l_i/2 - p + 1) k_i \sigma_i < v(A_i) \leq \mu_i - (l_i/2 - p) k_i \sigma_i & \text{if } 1 < p < l_i/2 \\ \mu_i - (l_i/2 - p + 1) k_i \sigma_i < v(A_i) \leq \mu_i & \text{if } p = l_i/2 \\ \mu_i < v(A_i) \leq \mu_i + (p - l_i/2) k_i \sigma_i & \text{if } p = l_i/2 + 1 \\ \mu_i + (p - l_i/2 - 1) k_i \sigma_i < v(A_i) \leq \mu_i + (p - l_i/2) k_i \sigma_i & \text{if } l_i/2 + 1 < p < l_i \\ \mu_i + (p - l_i/2 - 1) k_i \sigma_i < v(A_i) & \text{if } p = l_i \end{cases} \tag{3}$$

for an even l_i. In Figure 1 (top) we graphically represented the transformation for a given μ_i, σ_i, and k_i, and for the odd case with $l_i = 5$. Intuitively, k_i represents how far an endpoint of a bin is from the mean (μ_i) in terms of standard



Solution: $\boxed{k_1}\boxed{l_1}\boxed{\ldots}\boxed{k_i}\boxed{l_i}\boxed{\ldots}\boxed{k_n}\boxed{l_n}$

Optimization parameters:
$k_i \in (0, 1] \subset \mathbb{R}$ displacement
$l_i \in [2, \ldots, maxLabels] \subset \mathbb{N}$ number of labels

Fig. 1. Graphical representation of the discretization function (2) with $l_i = 5$ for some variable (top) and its encoding in the evolutionary algorithm for all variables (bottom).

deviations (σ_i). Applying it to the original numerical data set clearly leads to a categorical one from which rules can be extracted:

$$\mathcal{A}' = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \gamma_{m1} & \gamma_{m2} & \cdots & \gamma_{mn} \end{bmatrix}$$

For variables following the normal (Gaussian) distribution, this algorithm is a simple generalization of the common equal-width binning (with a particular configuration for l_i and k_i); yet, as we shall see, for non-normally distributed variables our technique often returns more significant bins.

Binning as an optimization problem. Our purpose is to devise a good discretization of a numerical data set using our distribution-based technique, that is, to *learn* a good pair of values (k_i, l_i), for each numerical attribute A_i. Following the taxonomy that we have identified in Section 2, we design our discretization algorithm as a wrapper, where APRIORI is used as a black-box. Because of the nature of the classical implementation of APRIORI, support and confidence must be set beforehand; therefore, optimizing these parameters may be less meaningful. On the other hand, we are not just interested in precise rules, but also in meaningful ones w.r.t. to specific domain-application (e.g., particular situations). As pointed out in [3], discretizing a numerical data set entails a notion of interestingness. In particular, we claim that the less number of categories, assuming that there are at least two categories per attribute, the more interesting are the rules, because, to some extent, having less labels implies that it is easier to interpret the discovered rules. Consistently with this observation, let $\bar{x} = [(x_1^1, x_1^2), (x_2^1, x_2^2), \ldots, (x_n^1, x_n^2)]$ be a vector of decision variables, each one of which is pair that represents the displacement k_i and the number of bins l_i of

the attribute A_i (cfr. Figure 1, bottom). Let $L(\bar{x})$ denote the collection of all l_is of a given vector $\bar{x}$, and let $Maxbins(\bar{x})$ (resp., $Meanbins(x)$) be the maximum value (resp., the average) in $L(\bar{x})$. Also, let $Card(\bar{x})$ (resp., $Meanconf(\bar{x})$) be the number of rules (resp., the average confidence of the rules) that emerge from an application of APRIORI over a numerical data set with n attributes discretized using (2) and (3), for a given minimum support s and minimal confidence c. We can instantiate (1) with four possible combinations of objectives:

$$\begin{cases} \min Maxbins(\bar{x}) \\ \max Card(\bar{x}) \end{cases} \quad (4) \qquad \begin{cases} \min Meanbins(\bar{x}) \\ \max Card(\bar{x}) \end{cases} \quad (5)$$

$$\begin{cases} \min Maxbins(\bar{x}) \\ \max Meanconf(\bar{x}) \end{cases} \quad (6) \qquad \begin{cases} \min Meanbins(\bar{x}) \\ \max Meanconf(\bar{x}) \end{cases} \quad (7)$$

Observe that minimizing the number of labels goes in favour of *interpretable* rules, while having at least two (non-empty) labels per attribute goes in favour of having *meaningful* rules.

Implementation. *Multiobjective evolutionary algorithms* are known to be particularly suitable to perform multiobjective optimization, as they search for multiple optimal solutions in parallel. In this experiment we have chosen the well-known NSGA-II (Non-dominated Sorted Genetic Algorithm) [7] algorithm, which is available as open-source from the *jMetal* suite [8]. NGSA-II is an elitist Pareto-based multiobjective evolutionary algorithm that employs a strategy with a binary tournament selection and a rank-crowding better function, where the rank of an individual in a population is the non-domination level of the individual in the whole population. As black-box rule extraction method we used the class *Apriori* from WEKA. We have represented each solution with an array of pairs:

$$(k_1, l_1), (k_2, l_2), \ldots, (k_n, l_n)$$

in which, as in Figure 1 (bottom), $2 \leq l_i \leq maxLabels$, where $maxLabels$ is a predetermined parameter, and $k_i \in (0, 1]$. The *initial population* is generated randomly. *Mutation* and *Crossover* were slightly modified to take into account the particular nature of our solutions. In particular, a mutation is performed, randomly with the same probability, over the first or the second component of the chosen attribute; the displacement is either substituted by random element in $(0, 1]$ or incremented or subtracted by the value 0.1, while the number of labels is either substituted by a random (integer) element in $[2, maxLabels]$ or incremented or subtracted by 1. Similarly, a crossover is performed at the level of the pair (k_i, l_i), that is, given two individuals to be crossed, two random pairs are chosen for each one of them and cross-exchanged. This particular configuration for performing both Mutation and Crossover is one among many others; the systematic study of other configurations is beyond the scope of this paper.

Variable	Unit	Mean	St.Dev.	Min	Median	Max
Air temp. (a)	$^{\circ}C$	10.9	8.4	-15.7	10.1	37.7
Solar dur. (d)	h	0.23	0.38	0	0	1
Wind speed (w)	ms^{-1}	3.13	1.95	0	3.00	19
% Rel. hum. (r)	–	74.9	17.3	20	79.0	100
Air pressure (p)	hPa	1003	8.5	906	1003	1028
Traffic (t)	–	2771	1795.0	30	3178	6713
NO_2	μgm^{-3}	50.4	23.2	1.7	49.4	231.6
NO_x	μgm^{-3}	142.2	103.7	3.9	123.7	1728.0

Table 1. Descriptive statistics of the data.

4 Data and Experimental Setup

Data origin. There is only one communication station for measuring the air quality in the city of Wrocław, and it is located within a wide street with two traffic lanes in each direction (GPS coordinates: 51.086390 North, 17.012076 East). The center of one of the largest intersections in the city with 14 traffic lanes is located approximately 30 meters from the measuring station, and is covered by traffic monitoring. The measurement station is located on the outskirts of the city, at $9.6km$ from the airport. *Pollution* data are collected by the Provincial Environment Protection Inspectorate and encompasses the hourly NO_2 and NO_x concentration values during full three years, from 2015 to 2017. *Traffic* data are provided by the Traffic Public Transport Management Department of the Roads and City Maintenance Board in Wrocław, and include hourly count of all types vehicles passing the intersection. Public meteorological data are provided by the Institute of Meteorology and Water Management, and they include: *air temperature*, *solar duration*, *wind speed*, *relative humidity*, and *air pressure*. In order to have uniform data, solar duration values have been re-normalized in the real interval $[0, 1]$ (as standard). The full data set contains 26304 observations. In the pre-processing phase, the instances with at least one missing value (617 samples, 2.3%) have been deleted. Some basic statistic indicators on the remaining 25687 instances are presented in Table 1, along with the symbol used in the tests for each variable. Unlike previous work in these data (e.g., in [11, 12]), we are interested in extracting rules for *particular cases*, not satisfactorily explained by a general regression model (e.g., sudden spikes in the contaminant concentrations, or unexpectedly low values of the contaminant concentration in presence of high values of the parameters that are linked to the common causes). In this experiment, we have focused on NO_2 values, eliminating the column NO_x from the data set; that is, our interest are the rules whose consequent is NO_2.

Experimental setup. We have performed one execution for each optimization model (from (4) to (7)) with population size 100 and 1000 total evaluations; our exploratory analysis showed that increasing the number of evaluations does not contribute to improve the results. We have set APRIORI to extract a maximum of 30 rules, with minimal support varying from 0.1 to 0.4 (with a 0.1 step) and

minimal confidence varying from 0.7 to 0.85 (with 0.05 step). Discovering at most 30 rules is purely arbitrarily, and may affect the outcome; nonetheless, since one of our goals is to extract more rules, such parameter suffices when increasing support and/or confidence. Moreover, observe that this choice depends on our particular application, as we are interested in predicting and explaining the amount of contaminant concentration (NO_2). Being multiobjective, each execution of a model gives rise to a Pareto front at the end of the last evolution to which we applied a simple decision making criteria, that is, we selected the best individual in terms of the second objective (number of rules in model (4) and (5), average confidence of the rules in model (6) and (7)).

5 Results and Discussion

Results. In Table 2 and Table 3 we show the results of the execution of optimization model (4) and (6). The results of model (5) (resp., (7)) are very similar to those of model (4) (resp., (6)), and therefore are not shown. Model (4) obviously extracts more rules than (6) in every configuration, given that we optimize precisely the number of rules. In each line of both tables, we show: the minimal support and confidence of rules with which APRIORI has been called; the number of rules of the selected final solution (more rules is better); the average length of the rules (smaller rules is better for, e.g., interpretability); the average confidence (more confident rules is better); and a *quality index* that depends on our particular application, that is, the number of rules whose consequent is NO_2 with highest values. In particular, we are interested in extracting rules that may possibly explain sudden, and not necessarily common, high values of the pollutant; yet, this parameter is merely informative, as not only depends on the application, but, also, its importance is subject to interpretation. As we can see, in both models we are able to extract progressively less rules as the minimal support and minimal confidence that are requested grow. To each solution is associated a *mask*, that is, the set of values $(k_1, l_1), (k_2, l_2), \ldots, (k_n, l_n)$ that correspond to the particular discretization; this mask, in turn, gives rise to the values that have been used to decide the bin to which each attribute of each particular instance belongs. As it often happens in rule extraction, interpreting the results is not immediate due to the number of extracted rules. While a correct and detailed NO_2 contamination model is outside the scope of this paper, from [11, 12] a general principle emerges: to more traffic, and to less wind, corresponds higher values of NO_2 in the atmosphere. Therefore, in searching for particular cases that may add explanatory value to the general model, we are interested in rules that defy such a general principle, and may therefore indicate that other parameters (inside or outside the available data set) play some role. There are three situations that may be interesting in this sense: higher NO_2 with lower traffic; higher NO_2 with no traffic (i.e., rules in which the traffic value has no role); and lower NO_2 with high traffic. As it turns out, from the optimization models (4) and (6), rules that belong to the third group do occur; examples of these are in Table 4. Observe that, since both optimization models

Min. Supp.	*Min. Conf.*	*# Rules*	*Av. length*	*Quality ind.*	*Av. Conf.*
0.1	0.70	30	2.96	9	0.76
	0.75	30	2.40	0	0.82
	0.80	29	2.79	0	0.83
	0.85	9	2.44	0	0.88
0.2	0.70	19	1.68	0	0.76
	0.75	9	1.77	0	0.80
	0.80	5	1.88	0	0.83
	0.85	1	2.00	0	0.86
0.3	0.70	4	1	0	0.74
	0.75	2	1.5	0	0.79
	0.80	1	2	0	0.81
	0.85	0	0	0	-
0.4	0.70	1	2	0	0.71
	0.75	1	2	0	0.79
	0.80	0	0	0	-
	0.85	0	0	0	-

Table 2. Results of the execution of the optimization model (4).

Min. Supp.	*Min. Conf.*	*# Rules*	*Av. length*	*Quality ind.*	*Av. Conf.*
0.1	0.70	14	2.57	1	0.88
	0.75	4	2	0	0.89
	0.80	2	2.5	0	0.9
	0.85	1	2	0	0.92
0.2	0.70	2	1.5	0	0.85
	0.75	2	1.5	0	0.85
	0.80	2	1.5	0	0.85
	0.85	1	2	0	0.86
0.3	0.70	1	1	0	0.79
	0.75	1	1	0	0.82
	0.80	1	2	0	0.82
	0.85	0	0	0	-
0.4	0.70	1	2	0	0.79
	0.75	1	2	0	0.8
	0.80	0	0	0	-
	0.85	0	0	0	-

Table 3. Results of the execution of the optimization model (6).

minimize the number of bins, such discovered rules encompass binary decisions (i.e., two labels) for all variables which are (arguably) informative for the specific domain-application.

Comparison with existing methods. A natural question at this point is whether we could have obtained similar results with the existing methods. To answer this question we considered a static discretizer (the class *Discretize* in WEKA, which behaves identically to the other available static filters in R

Rule	# of cases	*Conf.*
$d > 0.25 \wedge w > 3.18 \wedge h < 61.54 \wedge t > 2778.70 \rightarrow NO_2 \leq 79.40$	929	0.88
$w > 3.18 \wedge t > 2778.70 \rightarrow NO_2 \leq 79.40$	1596	0.86
$w > 3.18 \wedge t > 4403.88 \rightarrow NO_2 \leq 79.40$	888	0.92
$w > 3.18 \wedge h \leq 61.54 \wedge t > 2778.70 \rightarrow NO_2 \leq 79.40$	832	0.86

Table 4. Some interesting extracted rules.

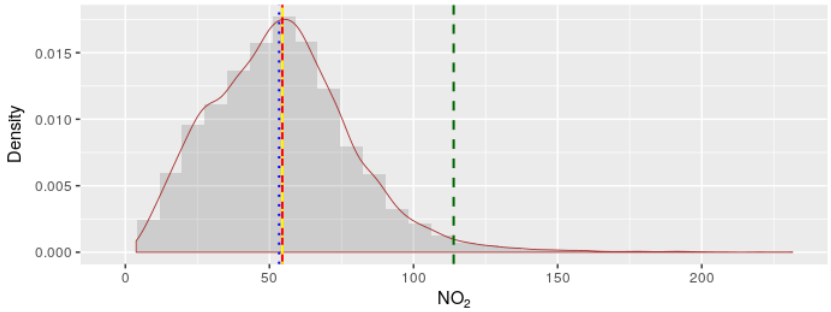


Fig. 2. Discretizing NO_2 in 2 bins. In red, we show the mean, in yellow we show the splitting points chosen by our optimization algorithm, in blue the splitting points obtained by static equal-frequency binning, and in green the splitting points obtained by static equal-width binning.

and in Python). As we have discussed in Section 2, a static discretizer usually offers equal-width and equal-frequency binning. While for normally distributed variables (i.e., Gaussian-like) both settings may return meaningful results, for non-normally distributed ones a further analysis of this aspect is necessary. Consider the case, for example, of the variable NO_2 in our data, which is strongly right-skewed. In Figure 2 we compare the result obtained by the three different solutions (equal-width, equal-frequency, and our dynamic discretizer) in the case of two bins; because of the distribution, equal-width binning in this case returns a bin ($NO_2 \leq 117.64$) which is meaningless: as a matter of fact, all rules that can be extracted with the different settings (support varying from 0.1 to 0.4, confidence varying from 0.7 to 0.85) return rules with that particular bin as consequent, which is uninformative because it is true almost in every instance. Therefore, equal-width binning makes little sense in our data set. Equal-frequency binning returned better results. For an adequate comparison, for each experimental setup, we have chosen the best mask (e.g., $[(0.2, 2), (0.4, 2), (1.0, 3), (1.0, 2), (0.7, 5), (0.6, 5), (1.0, 2)]$) returned by our algorithm, and asked the static filter to use that particular choice as number of bins for each variable (observe that deciding the number of bins *is* a problem in static binning - obviously, choosing the best option goes towards the fairest comparison). The results of such an experiments are shown in Table 5: only a few rules emerged, and none of them falls in one of the *interesting* categories that

Min. Supp.	*Min. Conf.*	*# Rules*	*Av. length*	*Quality ind.*	*Av. Conf.*
0.1	0.70	19	2.73	7	0.77
	0.75	0	-	-	-
	0.80	5	2	0	0.88
	0.85	6	2.16	0	0.88
0.2	0.70	0	-	-	-
	0.75	0	-	-	-
	0.80	0	-	-	-
	0.85	0	-	-	-
0.3	0.70	0	-	-	-
	0.75	0	-	-	-
	0.80	0	-	-	-
	0.85	0	-	-	-
0.4	0.70	0	-	-	-
	0.75	0	-	-	-
	0.80	0	-	-	-
	0.85	0	-	-	-

Table 5. Results of the execution of the static discretizer.

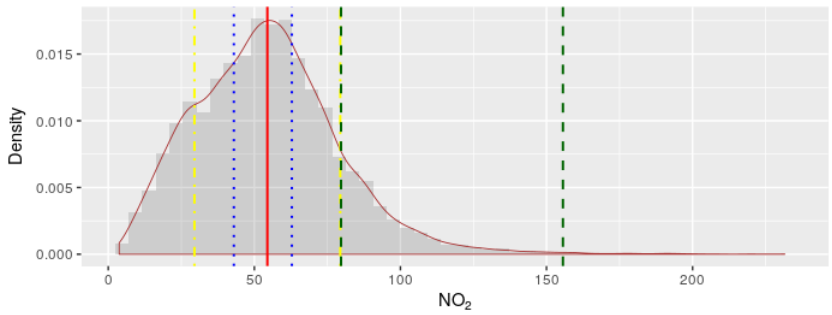


Fig. 3. Discretizing NO_2 in 3 bins. In red, we show the mean, in yellow we show the splitting points chosen by our optimization algorithm, in blue the splitting points obtained by static equal-frequency binning, and in green the splitting points obtained by static equal-width binning.

we have mentioned above. The most frequent result in all experiment shows a 3-bins splitting for NO_2, and Figure 3 gives us a hint of the problems that emerge during equal-frequency binning: as one can see, the resulting middle group is too narrow for a meaningful rule to be extracted.

6 Conclusions

In this paper we have considered a data set that represents the hourly measurement of several air quality parameters on a specific monitoring station in the center of Wrocław (Poland) during the interval of three years. Instead of

searching for a general explanation model for these data, we have focused on extracting meaningful local rules w.r.t. the pollutant, which may be useful to model the particular situations that occurred during the measurement period. We proposed and tested a multivariate, non-deterministic, distribution-based discretization algorithm coupled with the well-known rule extraction algorithm APRIORI, and implemented it via the evolutionary algorithm NSGA-II. We experimentally proved that our solution produces more rules and, in some cases, more interesting rules w.r.t. the specific domain-application than those produced by the discretization algorithms available in some open-source learning libraries.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proc. of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.
2. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of the 11th International Conference on Data Engineering*, pages 3–14, 1995.
3. R. Agrawal and R. Srikant. Mining quantitative association rules in large relational tables. *SIGMOD Rec.*, 25(2):1–12, 1996.
4. V. Bartík and J. Zendulka. *Distance-Based Methods for Association Rules Mining*, pages 689–694. IGI Global, 2nd edition, 2008.
5. B. Chlebus and S. Nguyen. On finding optimal discretizations for two attributes. In *Proc. of the 1st International Conference on Rough Sets and Current Trends in Computing*, pages 537–544, 1998.
6. Y. Collette and P. Siarry. *Multiobjective Optimization: Principles and Case Studies*. Springer Berlin Heidelberg, 2004.
7. K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, London, UK, 2001.
8. J. Durillo and A. Nebro. jMetal: a java framework for multi-objective optimization. *Avances in Engineering Software*, 42:760 – 771, 2011.
9. C. Emmanouilidis, A. Hunter, J. Macintyre, and C. Cox. A multi-objective genetic algorithm approach to feature selection in neural and fuzzy modeling. *Evolutionary Optimization*, 3(1):1–26, 2001.
10. F. Jiménez, G. Sánchez, J. García, G. Sciavicco, and L. Miralles. Multi-objective evolutionary feature selection for online sales forecasting. *Neurocomputing*, 234:75–92, 2017.
11. J. A. Kamińska. Probabilistic forecasting of nitrogen dioxide concentrations at an urban road intersection. *Sustainability*, 10(11), 2018.
12. J. A. Kamińska. A random forest partition model for predicting NO_2 concentrations from traffic flow and meteorological conditions. *Science of the Total Environment*, 651:475–483, 2019.
13. B. Lent, A. Swami, and J. Widom. Clustering association rules. In *Proc. of the 13th International Conference on Data Engineering*, pages 220–213. IEEE, 1997.
14. B. Minaei-Bidgoli, R. Barmaki, and M. Nasiri. Mining numerical association rules via multi-objective genetic algorithms. *Information Sciences*, 233:15–24, 2013.
15. M. N. Moreno, S. Segrera, V. F. López, and M. J. Polo. A method for mining quantitative association rules. In *Proc. of the 6th WSEAS International Conference on Simulation, Modelling and Optimization*, pages 173–178, 2006.

16. A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. C. Coello. A survey of multiobjective evolutionary algorithms for data mining: Part I. *IEEE Transactions on Evolutionary Computation*, 18(1):4–19, 2014.
17. B. Özden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. In *Proc. of the 14th International Conference on Data Engineering*, pages 412–421, 1998.
18. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
19. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.
20. A. Salleb-Aouissi, C. Vrain, C. Nortet, X. Kong, V. Rathod, and D. Cassard. Quantminer for mining quantitative association rules. *Journal of Machine Learning Research*, 14(1):3153–3157, 2013.
21. Q. Tong, B. Yan, and Y. Zhou. Mining quantitative association rules on overlapped intervals. In *Proc. of the 1st International Conference on Advanced Data Mining and Applications*, pages 43–50. Springer-Verlag, 2005.
22. I. Witten, E. Frank, and M. Hall. *Data mining: practical machine learning tools and techniques, 3rd Edition*. Morgan Kaufmann, Elsevier, 2011.

Appendix: NP-hardness of Optimal Binning

It is well known that optimal preprocessing is a computationally hard problem, and commonly accepted that learning algorithms use sub-optimal (deterministic or heuristic-based) solutions. Yet, precisely pinpointing the formal problem is mathematically interesting per se. Discretization as a classification task has been studied from the theoretical point of view in [5], and shown to be NP-hard; in this section we prove how discretization for rule extraction can be reduced to it.

Consider a numerical data set $\mathcal{A}$ with n features plus one (categorical) class D (in this context, referred to as *decision*):

$$\mathcal{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & d_1 \\ a_{21} & a_{22} & \dots & a_{2n} & d_s \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & d_m \end{bmatrix}$$

A pair (A_i, c), where A_i is an attribute and $c \in \mathbb{R}$ is called *cut*. Any set of cuts $\{(A_i, c_1), \dots, (A_i, c_l)\}$ implicitly defines the partition $\{(-\infty, c_1), [c_1, c_2), \dots, [c_l, +\infty)\}$ (denoted $\{\gamma_i^1, \dots, \gamma_i^{l_i}\}$ in Section 3). Having a partition for each attribute in A allows us to discretize the original data set to obtain a categorical one:

$$\mathcal{A} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \dots & \gamma_{1n} & d_1 \\ \gamma_{21} & \gamma_{22} & \dots & \gamma_{2n} & d_s \\ \dots & \dots & \dots & \dots & \dots \\ \gamma_{m1} & \gamma_{m2} & \dots & \gamma_{mn} & d_m \end{bmatrix}$$

Such a discretization is said to be *D-consistent* if there are no two instances with the same categorical values and different class; in other words a D-consistent transformation is the perfect classification. If $\mathcal{A}$ can be D-consistently discretized using at most l cuts (in total), we write $l \to \mathcal{A}$. Now, let $n = 2$ and let D be a binary class. One can define the language:

$$\mathcal{L} = \{\langle \mathcal{A}, l\rangle \mid l \to \mathcal{A}\}$$

In [5], it has been shown that deciding $\mathcal{L}$ is NP-hard. Adapting this result to our case is now easy. Indeed, the problem of finding the perfect discretization for extracting association rules from a numerical data set can be formalized as follows. Let $\mathcal{A}$ be a numerical data set with n attributes, each of which we can partition in the same way as before. Each discretization implicitly entails the existence of rules with a certain support and confidence, and, in particular, of rules whose consequent is any handpicked attribute; let us focus on rules with A_m as consequent. As we know, rules are evaluated by their support and confidence. Let us denote by $l, t \to^*_{s,c} \mathcal{A}$ the fact that $\mathcal{A}$ can be discretized in such a way that no more than l (total) bins are used, of which exactly t are the bins of the attribute A_m, in such a way that for each interval γ of the attribute A_m a group of rules can be extracted with confidence at least c and such that the sum of all supports is at least s. Observe that this discretization arguably describes what, in Section 3, we have called extracting the *highest possible number* of *meaningful* and *interpretable* rules. The problem of finding the perfect discretization in such terms can be then described as a decision problem, as follows:

$$\bar{\mathcal{L}} = \{\langle \mathcal{A}, l, t, s, c\rangle \mid l, t \to^*_{s,c} \mathcal{A}\}$$

Theorem 1. *$\bar{\mathcal{L}}$ is NP-hard.*

Proof. The proof is by polynomial reduction, that is, we prove that $\mathcal{L} \leq_p \bar{\mathcal{L}}$. To this end, let $\mathcal{A}$ be a numerical data set with $n = 2$ and a binary decision D. We produce a numerical data set $\mathcal{A}'$ with $n = 3$, where $A'_1 = A_1$, $A'_2 = A_2$, and A'_3 has domain $\{0, 1\}$ (that is, we copy the entire data set column-by-column, interpreting the decision as a numerical attribute). Now, we claim that $l \to \mathcal{A}$ if and only if $l+2, 2 \to^*_{1,1} \mathcal{A}'$, that is, $\langle \mathcal{A}, l\rangle \in \mathcal{L}$ if and only if $\langle \mathcal{A}, l+2, 2, 1, 1\rangle \in \bar{\mathcal{L}}$. Suppose, first, that $l + 2, 2 \to^*_{1,1} \mathcal{A}'$. By definition, there exists at least one rule $\gamma_1 \wedge \ldots \wedge \gamma_s \to 0$ and at least one rule $\gamma'_1 \wedge \ldots \wedge \gamma'_{s'} \to 1$ that hold in (the discretized counterpart of) $\mathcal{A}'$ with minimal confidence 1. Because of the requirement on the minimal total support, every instance is the support of some rule; because of the requirement on the minimal local confidence, no instance can be in the support of more than one rule. It is now immediate to interpret the set of extracted rules as a rule-based classifier with perfect score, that is, to see that $l \to \mathcal{A}$. Suppose, on the other hand, that $l \to \mathcal{A}$ holds. Every instance can be seen as a rule; all such rules have confidence at least one, because $\mathcal{A}$ is D-consistently discretized, and no instance is left without its corresponding rule, which means that the total support is 1, that is, that $l+2, 2 \to^*_{1,1} \mathcal{A}'$,as we wanted. □