16th World Congress of the International Fuzzy Systems Association (IFSA)
9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)

Reduction of Fuzzy Rule Bases Driven by the Coverage of Training Data

Michal Burda, Martin Štěpnička

Institute for Research and Applications of Fuzzy Modeling
Centre of Excellence IT4Innovations, University of Ostrava
30. dubna 22, 701 03 Ostrava, Czech Republic

Abstract

We present a technique for size reduction of a base of fuzzy association rules which is created using an automated approach and which is intended for inference. Our approach is based on controlling the coverage of training data by the rule base and removing only such rules that do not increase that coverage. Experiments show that such reduction is very effective while affecting the outputs of inference only very slightly.

Keywords: fuzzy association rules, fuzzy rule base, reduction, coverage

1. Introduction

Mining associations rules has become a very popular tool for its applicability as well as transparency. Let us recall that perhaps the first algorithm and related research no this topic was developed by Hájek et al. and it dates to late 60's, see [1, 2]. Although the original method called GUHA was being developed also later on [3], the major interest was attracted later in 90's due to the research of Agrawal, see e.g. [4, 5, 6].

Due to the natural interpretability of association rules, this method was very soon generalized for fuzzy association rules as well and developed in many directions including obtaining further extensions, e.g., visual tools such as fingrams. For some of the related works, we refer too [7, 8, 9, 10].

In this article, we address the problem of the reduction of generated fuzzy association rules. For the sake of readability, we will talk only about association rules as the classical (crisp) ones are only a special case of the fuzzy ones and thus, our research may be easily applied to classical association rules as well.

The research mainly aims at association rules that use antecedent and/or consequent fuzzy sets of an *inclusive type*, where some of the fuzzy sets may be included in other fuzzy sets. This does not mean that the results of the introduced research do not relate to a bit more "standard" rules[1] which often use a sort of partition (e.g. Ruspini partition [11]) of input and output axes and the inclusion of distinct antecedent and consequent fuzzy sets does not appear. However, in such cases, the reduction provided by the introduced algorithm will be significantly lower.

On the other hand, this fact does not decrease the applicability of the introduced investigation and its outputs at all. First, there is so-called *Perception-based Logical Deduction* [12, 13, 14] which is a specific inference method that is at disposal in the SW package LFLC [15] and also freely in the *lfl* R-package [16, 17]. This method deals with *evaluative linguistic expressions* [18]. These expressions stem from the basic trichotomy consisting of expressions `Small` (Sm), `Medium` (Me), `Big` (Bi) which are later on modified by linguistic hedges of narrowing or widening effect[2]. These hedges are of the inclusive type [19] and thus, the related fuzzy sets that model such expressions are also in an inclusion, see Fig 1.

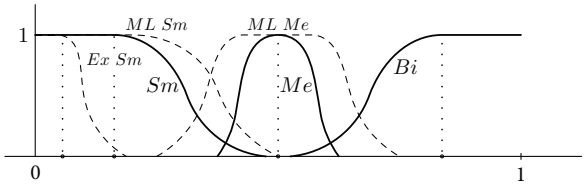

Figure 1: Fuzzy sets modeling some chosen evaluative linguistic expressions – demonstration of the inclusive type of the used linguistic hedges.

Of course, the model can be easily extended by adding further atomic expressions between those form the basic trichotomy, e.g., by adding `Lower Middle` and `Upper Middle` [14] as it is done in the above mentioned *lfl* R-package. The resulting algorithm provided in this article will be applicable also on such cases.

Second, there exist fuzzy rule base models that directly employ *at-least* and/or *at most modifiers* [19, 20], for the implicative models using fuzzy sets modified by such modifiers see e.g. [21, 22]. Of course, using expressions such as *at-least five* and *at-least seven* necessarily leads to the inclusion of the fuzzy sets that model these expressions.

[1] The word "standard" is understood from the engineering point of view.

[2] Typical hedges are Extremely (Ex), Very (Ve), More or Less (ML) or Roughly (Ro)

Finally, the above mentioned models using at-least and/or at-most modifiers can be easily generated from the standard ones, that use rather partitions of input and output domains, in order to group several antecedents and consequents together by the use of the modifiers. In other words, every standard model may be easily transformed to the one using inclusive expression and such a transformation is not an artificial step, but it may help to reduce the size of a rule base and to increase the readability.

2. Theoretical Background

Let $\mathcal{O} = \{o_1, \ldots, o_n\}$, $n > 0$, be a finite set of objects and $\mathcal{A} = \{a_1, \ldots, a_m\}$, $m > 0$ be a finite set of attributes (features). Each attribute can be considered as a fuzzy logical predicate: the truth value $a_i(o_j)$ would be from the interval $[0, 1]$ where the value $a_i(o_j)$ determines the degree of applicability of attribute a_i on object o_j, and boundary values 1 (resp. 0) denote that the i-th attribute fully applies (resp. does not apply) to object o_j.

For a subset $X \subseteq \mathcal{A}$ of fuzzy attributes, we define a new predicate of a logical conjunction by using a t-norm $\otimes$:

$$X(o_j) = \bigotimes_{a \in X} a(o_j), \quad j = 1, \ldots, n \tag{1}$$

and we put $X(o_j) = 1$ for $X = \emptyset$.

Recall that *t-norm* $\otimes$ is a binary operation on $[0, 1]$ that represents multiple-valued logical conjunction, i.e., a function $[0, 1] \times [0, 1] \to [0, 1]$ which is associative, commutative, increasing in both arguments and which satisfies the boundary condition $\alpha \otimes 0 = 0$ (and consequently also $\alpha \otimes 1 = \alpha$) for any $\alpha \in [0, 1]$ (c.f. [23]). Some well-known examples of t-norm are: the product t-norm $\alpha \otimes \beta = \alpha\beta$, the minimum t-norm $\alpha \otimes \beta = \min\{\alpha, \beta\}$, and the Łukasiewicz t-norm $\alpha \otimes \beta = \max\{0, \alpha + \beta - 1\}$.

An *association rule*[3] is a formula $X \rightharpoonup Y$, where $X \subset \mathcal{A}$ is an *antecedent*, $Y \subset \mathcal{A}$ is a *consequent*. Consider the following rule as an example:

$$\{\text{middle age}, \text{high education}\} \to \{\text{high income}\}.$$

Now, let us recall the most well-known quality measures of association rules, for details we refer to [2, 5, 8].

Definition 1. Let $\mathcal{A}$ be a finite set of attributes, $\mathcal{O}$ be a finite set of objects and let $X, Y \subset \mathcal{A}$. The quality measures *support* and *confidence* are defined as follows:

$$\text{supp}(X) = \frac{\sum_{o \in \mathcal{O}} X(o)}{|\mathcal{O}|}, \tag{2}$$

$$\text{supp}(X \rightharpoonup Y) = \text{supp}(X \cup Y), \tag{3}$$

$$\text{conf}(X \rightharpoonup Y) = \frac{\text{supp}(X \rightharpoonup Y)}{\text{supp}(X)}, \tag{4}$$

[3]We will consider only associations rules of implicative type. For the sake of readability, we omit this adjective and write simply association rule.

respectively.

Note that the definition of $\text{supp}(X \rightharpoonup Y)$ uses the union of antecedent and consequent. As the union only increases the set of attributes it actually has rather a conjunctive nature. Indeed, using formulas (1) and (2) we get:

$$\text{supp}(X \rightharpoonup Y) = \frac{\sum_{o \in \mathcal{O}} (X \cup Y)(o)}{|\mathcal{O}|}$$

where

$$(X \cup Y)(o) = \bigotimes_{a \in X \cup Y} a(o)$$

from which $\text{supp}(X \rightharpoonup Y) \leq \text{supp}(X)$ may be easily concluded for any $X, Y \in \mathcal{A}$.

Remark 1. *Definition 1 is mathematically correct but in order to use it in a meaningful way and in order to avoid getting trivial results, it is usually assumed that the antecedent and consequent are non-empty and disjoint, i.e., $X \neq \emptyset$, $Y \neq \emptyset$ and $X \cap Y = \emptyset$.*

Mining association rules basically consists in searching for all rules with support and confidence above some *minsupp* and *minconf* thresholds.

3. Rule Base Coverage of Data

Due to the *curse of dimensionality* phenomenon, the number of obtained association rules may turn to be enormous. This is usually strengthened in the case of fuzzy rules because a single numeric feature is often transformed to many fuzzy attributes with slightly different meaning, e.g. by applying linguistic expressions (such as "small", "medium", "big") combined with hedges (such as "very", "extremely", "rather" etc.). The obtained association rule base may be used as a model of machine learning for classification or prediction. It is desirable for the rule base not to be too complex for the sake of its understandability and in order to mitigate the risk of overfitting. An appropriate setting of minimum confidence and support thresholds is useful but usually not sufficient and some redundancy removal or simplification and size reduction algorithms are necessary to be used.

If some rules are being removed in order to decrease the size of the rule base, an appropriate "measure" of the quality of the rule base would be highly desirable in order to quantify the change of descriptive power of the rule base. Therefore, we introduce a concept called *the rule base coverage of data* that expresses the amount of data entries, for which there exists a rule with an antecedent that models ("covers") the data.

Definition 2. Let $\mathcal{O} = \{o_1, \ldots, o_n\}$ be a set of objects, $\mathcal{A} = \{a_1, \ldots, a_m\}$ be a set of attributes, and let $\mathcal{R} = \{R_1, \ldots, R_k\}$ be a set of association rules $R_i = A_i \rightharpoonup C_i$ with $A_i \subseteq \mathcal{A}$, $C_i \subseteq \mathcal{A}$, $i \in \{1, \ldots, k\}$.

Then the *coverage of a data $\mathcal{O}$ by rule base $\mathcal{R}$* is given as follows:

$$\mathrm{cov}_{\mathcal{O}}(\mathcal{R}) = \frac{1}{n}\sum_{j=1}^{n}\bigvee_{i=1}^{k} A_i(o_j)\ . \tag{5}$$

If there will be no danger of confusion and the data $\mathcal{O}$ will be evident from the context, we will simply write $\mathrm{cov}(\mathcal{R})$ instead of $\mathrm{cov}_{\mathcal{O}}(\mathcal{R})$.

Proposition 1. *Let $\mathcal{O}$ be a fixed set of objects then the following holds:*

(a) $\mathrm{cov}(\mathcal{R}) \in [0,1]$*;*

(b) $\mathrm{cov}(\emptyset) = 0$*;*

(c) If $\mathcal{R}' \subseteq \mathcal{R}$ *then* $\mathrm{cov}(\mathcal{R}') \leq \mathrm{cov}(\mathcal{R})$*;*

(d) If $S \in \mathcal{R}$ *such that* $S = \emptyset \rightharpoonup C$ *then* $\mathrm{cov}(\mathcal{R}) = 1$*;*

(e) If $S, T \in \mathcal{R}$ *such that* $S = A \rightharpoonup C$*,* $T = B \rightharpoonup C$*, and* $A \subset B$*, then* $\mathrm{cov}(\mathcal{R} \setminus \{T\}) = \mathrm{cov}(\mathcal{R})$*;*

(f) Let $S = A \rightharpoonup C$ *then* $\mathrm{supp}(A) = \mathrm{cov}(\{S\})$.

Proof. The Proposition can be easily proved using the given formulas and their fundamental properties. □

The definition of coverage presented above expresses a very intuitive measure of coverage of data by the generated rules. It is not necessarily expected to be close to 1, as some data sets may be generated rather by a pure noise than by some statistically significant dependencies expressible by associations[4]. On the other hand, high values of coverage clearly express high coverage of the data by generated association rules. Therefore, the concept of coverage may be very useful in setting the appropriate parameters of support and confidence thresholds.

4. Reduction of Rule Bases

In the subsequent sections we develop a technique of a rule base size reduction that modifies an original yet too big rule base $\mathcal{R}$ into a smaller reduced $\mathcal{R}'$. In that task, observing the difference between $\mathrm{cov}(\mathcal{R})$ and $\mathrm{cov}(\mathcal{R}')$ is highly desirable. Indeed, significant change in the coverage would mean that the reduced rule base by far does not cover the data samples that were covered by the original rule base, which is not advisable.

Definition 3. Let $\mathcal{O}$ be a fixed set of objects, $\mathcal{R}, \mathcal{R}'$ be rule bases such that $\mathcal{R}' \subseteq \mathcal{R}$ and $\mathrm{cov}(\mathcal{R}) > 0$. Then the *reduction ratio* is defined as follows:

$$\mathrm{reduct}_{\mathcal{O}}(\mathcal{R}, \mathcal{R}') = \frac{\mathrm{cov}_{\mathcal{O}}(\mathcal{R}')}{\mathrm{cov}_{\mathcal{O}}(\mathcal{R})}.$$

[4]Huge rule base with coverage equal or close to 1 may rise the suspicion of overfit.

Again, we are going to omit the $\mathcal{O}$ symbols whenever there is no danger of confusion. Obviously, $\mathrm{reduct}(\mathcal{R}, \mathcal{R}') \in [0,1]$, $\mathrm{reduct}(\mathcal{R}, \mathcal{R}) = 1$, and $\mathrm{reduct}(\mathcal{R}, \emptyset) = 0$.

The problem of possible redundancy of fuzzy rules is well-known and has been studied by many authors, see [24, 25, 26]. Investigations presented so far provided mainly algorithms that may slightly change the output of modified rule bases in comparison to original ones, but are very efficient in size-reducing simplification. It is worth recalling, e.g., the study [27] that focuses on redundancies in Takagi-Sugeno fuzzy rules or another interesting investigation dealing with redundancy in Takagi-Sugeno models [28] that is based on merging of similar rules.

In [29, 30, 13], the authors focused on theoretically-based algorithm of detection and removal of redundant rules in linguistic descriptions (rule bases) connected to the perception-based logical deduction [12]. However, the results still contained too many rules for the rule base to be transparent and easily interpretable. Therefore, in this paper we develop a size-reducing technique that reduces the rule base size even more. We relax from the requirement of not-changing the resulting function of the rule base, however, we try to ensure that the obtained modification will be as small as possible. Simultaneously, we keep the reduction ratio close to 1 in order to ensure that the coverage of the data does not change significantly, either.

Natural yet partly naive approaches would harm this goal. For example, an algorithm taking k rules with the highest confidence would not be very appropriate as the highest confidence is always obtained for rules with narrow antecedents and very wide consequents. Thus, such modified $\mathcal{R}'$ would not necessarily keep $\mathrm{cov}(\mathcal{R}')$ as close to $\mathrm{cov}(\mathcal{R})$ as possible and moreover, the inference on such modified rule base would change significantly its behaviour. The reason is that the wider the consequents, the closer the defuzzified output is to the middle of the output universe. A similar approach based on k rules with the highest support would not be appropriate as there is always a problem of the determination of the parameter k.

Another intuitive but not very fruitful approach is to use an algorithm that creates a reduced rule base $\mathcal{R}'$ from original $\mathcal{R}$ by selecting a rule with the highest support and then iteratively adding such rules that most increase the rule base coverage up to a certain threshold $\rho \in [0,1]$, i.e., until $\rho \leq \mathrm{reduct}(\mathcal{R}, \mathcal{R}')$. Setting the threshold ρ is elegant as it neither depends on the number of rules in the original rule base, nor on supports or confidences of the rules, and one transparently expresses the maximal allowed decrease of rule base coverage.

Unfortunately, this algorithm is not suitable, either. If the original rule base $\mathcal{R}$ contains rule R with empty antecedent, i.e., $R = \emptyset \rightharpoonup B$, then all details

about changed behaviour in exceptional cases, that may be modelled with other rules in $\mathcal{R}$ having different consequents, get lost because the trivially reduced rule base $\mathcal{R}' = \{R\}$ would have $\mathrm{cov}(\mathcal{R}') = 1$ and therefore also $\mathrm{reduct}(\mathcal{R}, \mathcal{R}') = 1$, see Proposition 1 for details. Similar lost occurs even if $\mathcal{R}$ contains a rule with antecedent having only few very broad predicates, i.e., predicates that are "almost-always" true for any data sample $o \in \mathcal{O}$ or most of such data samples.

4.1. The Reduction Algorithm

Our algorithm is driven by the threshold of allowed decrease of coverage too, but a care is taken on assuring the same diversity of rule consequents in reduced rule as in original rule base, see Algorithm 1.

Let $\rho \in [0, 1]$ be a reduction ratio threshold. First of all, the original rule base $\mathcal{R}$ is split into seperate sub-bases $\mathcal{R}_C$ containing only the rules with the same consequent C. Then these sub-bases are processed separately. The reduced rule base ${\mathcal{R}'}_C$ is created iteratively by adding rules that mostly increase coverage until $\mathrm{cov}({\mathcal{R}'}_C) < \mathrm{cov}(\mathcal{R}_C) \cdot \rho$. Finally, the reduced rule bases ${\mathcal{R}'}_C$ are joined into a single final rule base $\mathcal{R}'$:

$$\mathcal{R}' = \bigcup_{\forall C} {\mathcal{R}'}_C.$$

Obviously $\mathrm{cov}(\mathcal{R}') \geq \mathrm{cov}(\mathcal{R}) \cdot \rho$ and thus the reduction ratio $\mathrm{reduct}(\mathcal{R}, \mathcal{R}')$ is above ρ. Moreover, the same holds about all sub-bases, i.e., for any sub-base of rules with a single fixed consequent $\mathcal{R}_C$ and its related reduced sub-base ${\mathcal{R}'}_C$, the reduction ratio $\mathrm{reduct}(\mathcal{R}_C, {\mathcal{R}'}_C)$ is also above ρ. This ensures that all consequents are preserved in the reduced rule base for sufficiently (determined by ρ) many object samples compared to the original rule base.

Algorithm 1 uses the so-called *hill-climbing* heuristic: at each iteration, a rule with the potential to most increase the coverage is selected. Such method ensures a quite fast approach to obtain local minimum – a minimal rule base for which $\mathrm{cov}({\mathcal{R}'}_C) \geq \mathrm{cov}(\mathcal{R}_C) \cdot \rho$.

The performance of the algorithm is influenced by the order of rules being tried to add to ${\mathcal{R}'}_C$. Some rules have a great potential to increase the whole coverage and some others do not. We define rule R potential P_R as the biggest known amount of increase of coverage that the rule may cause by adding it to the reduced rule base. In the beginning, when ${\mathcal{R}'}_C$ is empty, each rule $R = A \rightharpoonup B$ has potential $P_R = \mathrm{supp}(A)$.

During construction of reduced rule base ${\mathcal{R}'}_C$, we take a rule R with greatest potential P_R and compute $\mathrm{cov}({\mathcal{R}'}_c \cup \{R\})$. Then we update its potential P_R to $P_R = \mathrm{cov}({\mathcal{R}'}_c \cup \{R\}) - \mathrm{cov}({\mathcal{R}'}_c)$ If such rule (in some of the next steps) occurs again as a rule with greatest potential, we can be sure that it is a rule with the greatest potential, and thus, it is the rule that increases the coverage most.

For selection of the rules by greatest potential, we use a priority queue that sorts the rules internally by their potential (in descending order) and then also by *timestamp* (in ascending order). Timestamp is a small trick to prevent miss-selection if two rules occur in the queue with the same potential. Then that rule is selected whose potential is "older" i.e. rule that was in the queue stored with that potential earlier.

For $k = |\mathcal{R}|$, the time complexity of the inner while-loop is of the complexity order $O(k)$, then the time complexity of the whole algorithm is of the complexity order $O(k^2)$. However, in average, both while-loops iterate k-times very seldom. In fact, the best rule to add is found quite quickly after relatively few iterations as can be seen from the experiment introduced Section 5. The space complexity is linear, $O(k)$.

One may see the fact that our algorithm finds only local minima as a drawback. To vindicate our approach please note that finding a global minimum – a smallest rule base possible – is an NP-hard problem. Since the reduction of rule base size is quite huge even in local minimum, we see our approach as rational compromise between time complexity and the quality of the solution.

Also note that the outer for-loop can be executed concurrently and hence the algorithm is easily parallelizable for computer architectures with up to p CPUs, where p is the number of unique consequents in the input rule base $\mathcal{R}$.

4.2. Pre-processing During the Rule Base Construction

A broadly accepted technique for mining of association rules [4, 6] is searching for rules starting from empty antecedents and adding new predicates to them until the rule support falls below the user-specified minimum support threshold. If a rule $A \rightharpoonup C$ has a support below some minimum threshold then any rule $B \rightharpoonup C$ such that $B \supset A$ would necessarily have the support below that minimum too. Such implication is known as the *Apriori condition*.

Accordingly to Apriori condition, if a rule has a support below given threshold, adding new predicates to its antecedent is not needed because no new rule can be created with sufficiently large support.

Similarly, if $R = A \rightharpoonup C$ is a found rule, $R \in \mathcal{R}$, we can stop searching for rules $R' = B \rightharpoonup C$ such that $B \supset A$ (i.e. with a longer antecedent) because according to Proposition 1, $\mathrm{cov}(\mathcal{R} \cup \{R'\}) = \mathrm{cov}(\mathcal{R})$ and hence adding R' to the rule base would not increase the coverage at all.

As can be seen from Table 1, such pruning seems to be very efficient. Table 1 shows number of rules found without pruning (column 5) and with pruning (column 8). One can see the size of rule base is very reduced even without the reduction algorithm

Algorithm 1 Reduction of a rule base driven by the rule base coverage ratio

Inputs: set of objects $\mathcal{O}$, rule base $\mathcal{R}$, reduction ratio threshold ρ
Output: reduced rule base $\mathcal{R}'$

```
1: function REDUCE(R, O, ρ)
2:     R' ← ∅
3:     for each unique consequent C of any rule from R do
4:         R_C ← (R ∈ R | C is the consequent of R)
5:         R'_C ← ∅
6:         bestRule ← ∅
7:         bestCoverage ← 0
8:         timestamp ← 0
9:         queue ← R_C (queue sorts the rules by potential P_R (decreasing) and timestamp T_R (increasing))
10:        while cov_O(R'_C) < cov_O(R_C) · ρ do
11:            while queue.top ≠ bestRule do
12:                timestamp ← timestamp + 1
13:                R ← remove top rule from the queue
14:                P_R ← cov_O(R'_C ∪ {R}) − cov_O(R'_C)
15:                T_R ← timestamp
16:                push R into the queue
17:                if bestCoverage < cov_O(R'_c ∪ {R}) then
18:                    bestCoverage ← cov_O(R'_c ∪ {R})
19:                    bestRule ← R
20:                end if
21:            end while
22:            remove top rule from the queue
23:            R'_C ← R'_C ∪ {bestRule}
24:            bestRule ← ∅
25:            bestCoverage ← 0
26:        end while
27:        R' ← R' ∪ R'_C
28:    end for
29:    return R'
30: end function
```

being applied. After application of the reduction algorithm proposed in Section 4.1, the size of rule base decreases even more. Moreover, the performance of reduction is much better than without pruning (compare column 7 with column 10).

The proposed pruning condition speeds-up both the rule search algorithm (because less combinations has to be evaluated) and the rule base reduction algorithm (as the smaller original rule base has to be processed).

5. The Experiment

We have executed two experiments to assess the usefulness of reduction. In the first experiment, the performance of the algorithm is evaluated from the perspective of execution time and the amount of reduction. In the second experiment, a quality of the reducted rule base is evaluated in the sense of the quality of prediction by a logical deduction.

5.1. Reduction Performance

To evaluate the execution time and the ability to reduce the size of rule bases, the proposed algorithm was tested on several data sets from the UCI Machine Learning Repository [31].

The tests were executed on Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz with 16 GB RAM running the Ubuntu 14.04 operating system with kernel Linux version 3.13.0-43-generic. The algorithm was written in C++ and evaluated in the environment of the R software version 3.1.2. The fuzzy association rules search algorithm was taken from the *lfl* (Linguistic Fuzzy Logic) package [16, 17].

Single data column was selected as consequent, the rest were antecedents. Each categorical column was transformed into c fuzzy sets (where c is the number of categories), such that each set represents a different category and objects have trivial membership degrees $\{0, 1\}$ only. Each numeric column was transformed into 16 different fuzzy sets by applying linguistic expressions (small, medium big) and linguistic hedges to the original value. Rows with missing values were omitted. Association rules were searched with minimum support threshold = 0.05, minimum confidence threshold = 0.7, and maximum size of the antecedent (i.e. $|A|$ for rule $A \rightharpoonup C$) = 4. The reduction ratio was set

Data	Rows	LHS	RHS	Without Pre-processing			With Pre-processing			Reduced
				Rules	Scans	Time	Rules	Scans	Time	Rules
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
abalone	4177	115	16	12338	17803	1.462	167	336	0.031	32
bikesharing	731	208	16	78861	128297	3.825	2922	7632	0.166	93
german	1000	166	16	119430	5	3.818	5	5	0.005	5
heart	270	208	16	296925	476060	12.684	3805	12900	0.157	120
housing	506	208	16	73361	83181	3.519	1281	3603	0.070	85
skillcraft1	3395	304	16	44549	102816	5.108	3621	10526	0.396	36
winequality	4898	176	16	50953	2	2.578	2	2	0.024	2

Table 1: Evaluation of the reduction algorithm on several datasets from the UCI repository [31]. The columns from the left are: 1. data set name, 2. number of rows, 3. number of (fuzzy) sets in LHS (antecedent), 4. number of (fuzzy) sets in RHS (consequent); 5. & 8. size of rule base before reduction, 6. & 9. number of scans of dataset during reduction (without initial scan to determine original rule base coverage), 7. & 10. reduction execution time (in seconds): first without and next with pre-processing pruning condition; 11. number of rules after reduction.

Data	Original Size	Reduction Threshold								
	(without pre-proc.)	1.00	0.99	0.98	0.97	0.96	0.95	0.90	0.85	0.8
abalone	12338	70	32	27	23	21	18	13	10	9
bikesharing	78861	148	93	77	70	64	56	45	34	27
german	119430	5	5	5	5	5	5	5	5	5
heart	296925	175	120	110	95	85	75	55	40	30
housing	73361	128	85	72	65	58	54	38	30	22
skillcraft1	44549	155	36	27	21	17	16	10	8	7
winequality	50953	2	2	2	2	2	2	2	2	2

Table 2: Number of rules obtained for different levels of reduction threshold

Data	Original	Difference from Orig.	
		Pre-proc.	Reduced
abalone	3.453	−0.003	+0.061
bikesharing	1623.821	+3.439	−144.491
german	0.586	−0.046	−0.021
heart	0.586	0	−0.014
housing	7.546	+0.001	−0.039
skillcraft1	1.443	0	−0.007
winequality	0.882	0	0

Table 3: Evaluation of the prediction performance of the rule bases. Here RMSE is reported for original rule base, rule base created with pruning (pre-processing) as described in Section 4.2, and reduced rule base.

to 0.99. Table 1 shows summary information about data being used and the results obtained.

Although the theoretical complexity of the reduction algorithm is $O(k^2)$ (where $k = |\mathcal{R}|$) it can be seen that the number of scans of data set is no more than $4 \cdot k$ (k scans for initial determination of the coverage + $2k$ to $3k$ scans in average). The heuristic based on rule potential seems to be very efficient.

5.2. Rule Base Quality

To measure how the predictive power of a rule base changes after application of the reduction algorithm, the following experiment was set. The source data set was split into training ($\frac{3}{4}$) and testing ($\frac{1}{4}$) part. On training part, the rules were searched and reduction performed. The original rule base and the reduced rule base were then used for prediction of the target attribute value on the testing data set. As an inference mechanism, the Perception-based Logical Deduction [12] was used. The difference between predicted p and real value r was evaluated using the *Root Mean Square Error*:

$$\mathrm{RMSE}(p, r) = \sqrt{\frac{1}{q}\sum_{i=1}^{q}(p_i - r_i)^2}.$$

As can be seen from Table 3, the RMSE of prediction has no change or changes only very slightly for reduced rule base compared to original rule base. Most of the times, the error even decreases if using reduction on rule base. We believe that reduction helps eliminating overfit – too complex model such as large rule base may capture also the noise.

6. Conclusion

We have proposed a novel technique for size reduction of rule bases. The reduction is driven by the coverage of data by the rules so that the coverage must not decrease below a user-specified threshold. We have shown the performance of the proposed algorithm, the effectiveness of reduction, and we have also demonstrated that the resulting reduced rule

base does not produce worse predictions if used with the Perception-based Logical Deduction.

The reduction algorithm together with a tool for searching of fuzzy association rules in data as well as the Perception-based Logical Deduction is programmed as an open-source package, *lfl* [16, 17], for the R statistical envirnoment that runs on Windows, Linux, or Macintosh systems.

Acknowledgement

This work was supported by the European Regional Development Fund in the project of IT4Innovations Centre of Excellence (CZ.1.05/1.1.00/02.0070, VP6).

References

[1] P. Hájek, I. Havel, and M. Chytil. The GUHA method of automatic hypotheses determination. In *Computing 1*, pages 293–308, 1966.

[2] P. Hájek. The question of a general concept of the GUHA method. *Kybernetika*, 4:505–515, 1968.

[3] P. Hájek and T. Havránek. *Mechanizing hypothesis formation: Mathematical foundations for a general theory.* Springer-Verlag, Berlin/Heidelberg/New York, 1978.

[4] R. Agrawal, T. Imielinski, and A. Swami. Mining associations between sets of items in massive databases. In *ACM SIGMOD 1993 Int. Conference on Management of Data*, pages 207–216, Washington D.C., 1993.

[5] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. on Very Large Databases*, pages 487–499, Chile, 1994. AAAI Press.

[6] R. Agrawal. Fast discovery of association rules. In *Advances in knowledge discovery and data mining*, pages 307–328. AAAI Press / MIT Press, 1996.

[7] T. Sudkamp. Examples, counterexamples, and measuring fuzzy associations. *Fuzzy Sets and Systems*, 149(1):57–71, 2005.

[8] P. Hájek, M. Holeňa, and J. Rauch. The GUHA method and its meaning for data mining. *Journal of Computer and Systems Sciences*, 76:34–48, 2010.

[9] D. P. Pancho, J. M. Alonso, J. Alcalá-Fdez, and L. Magdalena. Interpretability analysis of fuzzy association rules supported by fingrams. In *Proc. 8th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2013)*, Advances in Intelligent Systems Research, pages 469–474. Atlantic Press, 2013.

[10] M. Burda, P. Rusnok, and M. Štěpnička. Mining linguistic associations for emergent flood prediction adjustment. *Advances in Fuzzy Systems*, page DOI: 10.1155/2013/131875, 2013.

[11] E. H. Ruspini. A new approach to clustering. *Inf. Control*, 15:22–32, 1969.

[12] V. Novák. Perception-based logical deduction. In B. Reusch, editor, *Computational Intelligence, Theory and Applications*, Advances in Soft Computing, pages 237–250, Berlin, 2005. Springer.

[13] A. Dvořák, M. Štěpnička, and L. Štěpničková. On redundancies in systems of fuzzy/linguistic if-then rules under perception-based logical deduction inference. *Fuzzy Sets and Systems*, in press.

[14] A. Dvořák, M. Štěpnička, and P. Vitásková. On perception-based logical deduction and its variants. In *Proc. 16th World Congress of the International Fuzzy Systems Association and 9th Conference of the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT 2015)*, Advances in Intelligent Systems Research, page in this issue, Gijón, 2015. Atlantic Press.

[15] A. Dvořák, H. Habiballa, V. Novák, and V. Pavliska. The software package LFLC 2000 - its specificity, recent and perspective applications. *Computers in Industry*, 51:269–280, 2003.

[16] M. Burda. Linguistic fuzzy logic in R. In *IEEE International Conference on Fuzzy Systems*, Istanbul, 2015.

[17] M. Burda. *lfl: Linguistic Fuzzy Logic (R package on CRAN)*, 2015. http://cran.r-project.org/web/packages/lfl/ [2015-01-25].

[18] V. Novák. A comprehensive theory of trichotomous evaluative linguistic expressions. *Fuzzy Sets and Systems*, 159(22):2939–2969, 2008.

[19] M. De Cock and E.E. Kerre. Fuzzy modifiers based on fuzzy relations. *Information Sciences*, 160:173–199, 2004.

[20] U. Bodenhofer and P. Bauer. Interpretability of linguistic variables: A formal account. *Kybernetika*, 41(2):227–248, 2005.

[21] M. Štěpnička and B. De Baets. Implication-based models of monotone fuzzy rule bases. *Fuzzy Sets and Systems*, 232:134–155, 2013.

[22] M. Štěpnička and B. De Baets. Interpolativity of at-least and at-most models of monotone single-input single-output fuzzy rule bases. *Information Sciences*, 234:16–28, 2013.

[23] E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*, volume 8 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, 2000.

[24] R. Babuška and M. Setnes. Data-driven construction of transparent fuzzy models. In H.B. Verbruggen, H.-J. Zimmermann, and R. Babuška, editors, *Fuzzy Algorithms for Control*, pages 83–106. Kluwer, Boston, 1999.

[25] M. Setnes. Fuzzy rule base simplification using similarity measures. Master's thesis, Delft University of Technology, Defit, Netherlands, 1995.

[26] M. Setnes, V. Lacrose, and A. Titli. Com-

plexity reduction methods for fuzzy systems. In H.B. Verbruggen, H.-J. Zimmermann, and R. Babuška, editors, *Fuzzy Algorithms for Control*, pages 185–218. Kluwer, Boston, 1999.

[27] S. Galichet and L. Foulloy. Size reduction in fuzzy rulebases. In *Proc. IEEE International Conference On Systems, Man and Cybernetics*, pages 2107–2112, San Diego, 1998.

[28] E. Lughofer and E. Hüllermeier. On-line redundancy elimination in evolving fuzzy regression models using a fuzzy inclusion measure. In *Proc. of EUSFLAT-LFA 2011*, pages 380–387, Aix-les-Bains, July 2011.

[29] A. Dvořák, M. Štěpnička, and L. Vavříčková. Redundancies in systems of fuzzy/linguistic if-then rules. In *Proc. 7th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2011) and LFA-2011*, Advances in Intelligent Systems Research, pages 1022–1029, Paris, 2011. Atlantic Press.

[30] L. Štěpničková, M. Štěpnička, and A. Dvořák. New results on redundancies of fuzzy/linguistic if-then rules. In *Proc. 8th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2013)*, pages 400–407, Milano, 2013. Atlantic Press.

[31] A. Asuncion and D.J. Newman. UCI machine learning repository, 2015. http://www.ics.uci.edu/~mlearn/MLRepository.html [2015-01-15].